(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 523 544 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **92111660.4**

(51) Int. Cl.[5]: **G06F 15/324**

(22) Date of filing: **09.07.92**

(30) Priority: **12.07.91 JP 172168/91**
15.07.91 JP 173616/91

(43) Date of publication of application:
20.01.93 Bulletin 93/03

(84) Designated Contracting States:
DE FR GB

(71) Applicant: **MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.**
1006, Oaza Kadoma
Kadoma-shi, Osaka-fu, 571(JP)

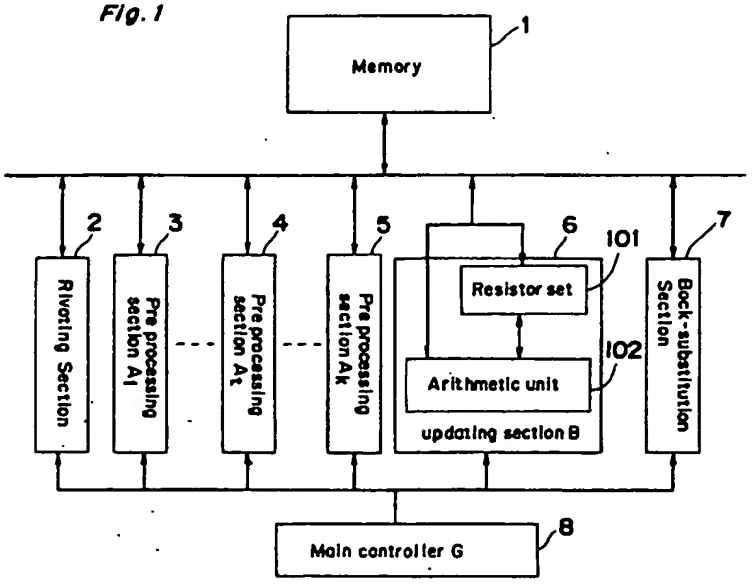
(72) Inventor: **Mochizuki, Yoshiyuki**
3-9-3 Sagisu, Fukushima-ku
Osaka-shi, Osaka(JP)

(74) Representative: **Eisenführ, Speiser & Partner**
Martinistrasse 24
W-2800 Bremen 1(DE)

(54) Calculating equipment for solving systems of linear equation.

(57) A linear calculating equipment comprises a memory for storing a coefficient matrix, a known vector and an unknown vector of a given system of linear equations, a pivoting device for choosing pivots of the matrix, a plurality of preprocessors for executing K steps of preprocessing for multi-pivot simultaneous elimination, an updating device for updating the elements of the matrix and the components of the vectors, a register set for storing values of the variables, a back-substitution device for obtaining a solution and a main controller for controlling the linear calculating equipment as a whole.

Fig. 1



EP 0 523 544 A2

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to calculating equipment for solving systems of linear equations, parallel calculating equipment for solving systems of linear equations, and methods of parallel computation for solving systems of linear equations.

2. Description of the Related Art

The need for solving systems of linear equations at high speed frequently arises in numerical analysis of the finite element method and the boundary element method and other processes of technical calculation.

Among algorithms based on direct methods of solving systems of linear equations is Gauss elimination method based on bi-pivot simultaneous elimination, which is described in Takeo Murata, Chikara Okuni and Yukihiko Karaki, "Super Computer--Application to Science and Technology," Maruzen, 1985. pp 95-96. The bi-pivot simultaneous elimination algorithm eliminates two columns at the same time by choosing two pivots at one step. It limits simultaneous elimination to two columns and the choice of pivots to partial pivoting by row interchanges. Furthermore it considers the speeding up of its process in terms of numbers of repetition of do-loops only.

If simultaneous elimination is not limited to two columns and extended to more than two columns, the corresponding algorithms will be hereafter called multi-pivot simultaneous elimination algorithms.

A similar algorithm to multi-pivot simultaneous elimination algorithms is described in Jim Armstrong, "Algorithm and Performance Notes for Block LU Factorization," International Conference on Parallel Processing, 1988, Vol. 3, pp 161-164. It is a block LU factorization algorithm intended to speed up matrix operations and should be implemented in vector computers or computers with a few multiplexed processors.

Therefore, according to prior art, there has not yet been developed Gauss elimination method or Gauss-Jordan elimination method which is based on multi-pivot simultaneous elimination and can be efficiently implemented in scalar computers and parallel computers.

SUMMARY OF THE INVENTION

The object of the present invention is therefore to provide high-speed parallel calculating equipment and methods of parallel computation for solving systems of linear equations by means of Gauss elimination method and Gauss-Jordan's method based on multi-pivot simultaneous elimination.

In order to achieve the aforementioned objective, according to one aspect of the present invention, there are provided

a memory that stores reduced coefficient matrices $A^{(r)}$ with zeroes generated from the first to the r-th column and corresponding known vectors $b^{(r)}$ and an unknown vector x expressed by

$$A^{(r)} = (a_{ij}^{(r)}) \ , \ 1 \le i, j \le n,$$

$$b^{(r)} = (b_1^{(r)}, b_2^{(r)}, \ . \ . \ . \ , b_n^{(r)})^t, \qquad (1)$$

$$x = (x_1, x_2, \ . \ . \ . \ , x_n)^t$$

for a given system of linear equations

$$A^{(0)}x = b^{(0)}. \qquad (2)$$

a pivot choosing section that is connected to the memory, chooses a pivot in the i-th row of $A^{(i-1)}$, and interchanges the i-th column with the chosen pivotal column,

a preprocessing section A_1 that, immediately after the pivot choosing section's above operation determines the transposed pivot

$$a_{pk+1\,pk+1}^{(pk)}, \tag{3}$$

calculates

$$a_{pk+1\,j}^{(pk+1)} = a_{pk+1\,j}^{(pk)} / a_{pk+1\,pk+1}^{(pk)} \tag{4}$$

for $pk + 2 \leq j \leq n$ and

$$b_{pk+1}^{(pk+1)} = b_{pk+1}^{(pk)} / a_{pk+1\,pk+1}^{(pk)}, \tag{5}$$

k - 1 preprocessing sections A_t, where $t = 2, 3, \ldots, k$, each of which is connected to the memory and calculates

$$Reg_{pk+t}^{(0)} = a_{pk+t\,pk+1}^{(pk)}, \tag{6}$$

$$Reg_{pk+t}^{(1)} = a_{pk+t\,pk+2}^{(pk)} - Reg_{pk+t}^{(0)} a_{pk+1\,pk+2}^{(pk+1)}, \tag{7}$$

$$\ldots,$$

$$Reg_{pk+t}^{(t-2)} = a_{pk+t\,pk+t-1}^{(pk)} - \sum_{m=1}^{t-2} Reg_{pk+t}^{(m-1)} a_{pk+m\,pk+t-1}^{(pk+m)}, \tag{8}$$

$$a_{pk+t\,j}^{(pk+t-1)} = a_{pk+t\,j}^{(pk)} - \sum_{m=1}^{t-1} Reg_{pk+t}^{(m-1)} a_{pk+m\,j}^{(pk+m)}, \tag{9}$$

$$b_{pk+t}^{(pk+t-1)} = b_{pk+t}^{(pk)} - \sum_{m=1}^{t-1} Reg_{pk+t}^{(m-1)} b_{pk+m}^{(pk+m)} \tag{10}$$

for $pk + t \leq j \leq n$, and, immediately after the pivot choosing section determines the transposed pivot

$$a_{pk+t\,pk+t}^{(pk+t-1)}, \tag{11}$$

calculates

$$a_{pk+tj}^{(pk+t)} = a_{pk+tj}^{(pk+t-1)} / a_{pk+tpk+t}^{(pk+t-1)}, \quad (12)$$

$$b_{pk+t}^{(pk+t)} = b_{pk+t}^{(pk+t-1)} / a_{pk+tpk+t}^{(pk+t-1)} \quad (13)$$

for $pk + t + 1 \leq j \leq n$,

an updating section B that is connected to the memory, comprises a set of k registers and an arithmetic unit, and calculates

$$Reg_i^{(0)} = a_{ipk+1}^{(pk)}, \quad (14)$$

$$Reg_i^{(1)} = a_{ipk+2}^{(pk)} - Reg_i^{(0)} a_{pk+1pk+2}^{(pk+1)}, \quad (15)$$

. . . . .

$$Reg_i^{(k-1)} = a_{i(p+1)k}^{(pk)} - \sum_{m=1}^{k-1} Reg_i^{(m-1)} a_{pk+m(p+1)k}^{(pk+m)}, \quad (16)$$

$$a_{ij}^{((p+1)k)} = a_{ij}^{(pk)} - \sum_{m=1}^{k} Reg_i^{(m-1)} a_{pk+mj}^{(pk+m)}, \quad (17)$$

$$b_i^{((p+1)k)} = b_i^{(pk)} - \sum_{m=1}^{k} Reg_i^{(m-1)} b_{pk+m}^{(pk+m)} \quad (18)$$

for $(p + 1)k + 1 \leq i, j \leq n$ retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in the register set,

a back-substitution section that is connected to the memory and obtains the value of the unknown vector x by calculating

$$x_i = b_i^{(n)} \quad (19)$$

and

$$b_h^{(n+h-i+1)} = b_h^{(n+h-i)} - a_{hi}^{(h)} x_i \qquad (20)$$

for $1 \leq h \leq i - 1$ for $i = n, n - 1, \ldots, 1$ in this order of i, and

a main controller G that, if n is a multiple of k, instructs the pivot choosing section, the preprocessing sections $A_1, \ldots, A_k$, and the updating section B to repeat their above operations for $p = 0, 1, \ldots, n/k - 2$, and instructs the pivot choosing section and the preprocessing sections $A_1, \ldots, A_k$ to execute their above operations for $p = n/k - 1$, and, if n is not a multiple of k, instructs the pivot choosing section, the preprocessing sections $A_1, \ldots, A_k$, and the updating section B to repeat their above operations for $p = 0, 1, \ldots [n/k] - 1$, where [x] denotes the greatest integer equal or less than x, and instructs the pivot choosing section and the preprocessing sections $A_1, \ldots, A_{n-[n/k]k}$ to execute their above operations, and in both cases, instructs the back-substitution section to obtain the unknown vector x.

According to another aspect of the present invention there are provided

a memory that stores coefficient matrices $A^{(r)}$, known vectors $b^{(r)}$ and the unknown vector x expressed by (1) for a given system of linear equations (2),

a pivot choosing section that is connected to the memory, chooses a pivot in the i-th row of $A^{(i-1)}$, and interchanges the i-th column with the chosen pivotal column,

a preprocessing section A_1 that, immediately after the pivot choosing section's above operation determines the transposed pivot (3), calculates (4) for $pk + 2 \leq j \leq n$ and (5),

k - 1 preprocessing sections A_t, where $t = 2, 3, \ldots, k$, each of which is connected to the memory, calculates (6), (7), . . . , (10) for $pk + t \leq j \leq n$, and, immediately after the pivot choosing section determines the transposed pivot (11), calculates (12) and (13) for $pk + t + 1 \leq j \leq n$,

an updating section B' which is connected to the memory, comprises a set of k registers and an arithmetic unit, and calculates (14), (15), . . . , (18) for $1 \leq i \leq pk$, $(p + 1)k + 1 \leq i \leq n$, $(p + 1)k + 1 \leq j \leq n$ if n is a multiple of k or $p < [n/k]$ and for $1 \leq i \leq [n/k]k$, $[n/k]k + 1 \leq j \leq n$ otherwise, retaining the values of

$$Reg_1^{(0)}, \ldots, Reg_1^{(k)}$$

in the register set,

k - 1 postprocessing sections C_t, where $t = 1, 2, \ldots, k - 1$, each of which is connected to the memory and calculates

$$Reg^{(0)} = a_{pk+1\,pk+t+1}^{(pk+t)}, \tag{21}$$

$$Reg^{(1)} = a_{pk+2\,pk+t+1}^{(pk+t)}, \tag{22}$$

. . . ,

$$Reg^{(t-1)} = a_{pk+t\,pk+t+1}^{(pk+t)}, \tag{23}$$

$$a_{pk+1\,j}^{(pk+t+1)} = a_{pk+1\,j}^{(pk+t)} - Reg^{(0)}\, a_{pk+t+1\,j}^{(pk+t+1)}, \tag{24}$$

$$a_{pk+2\,j}^{(pk+t+1)} = a_{pk+2\,j}^{(pk+t)} - Reg^{(1)}\, a_{pk+t+1\,j}^{(pk+t+1)}, \tag{25}$$

. . . ,

$$a_{pk+t\,j}^{(pk+t+1)} = a_{pk+t\,j}^{(pk+t)} - Reg^{(t-1)}\, a_{pk+t+1\,j}^{(pk+t+1)}, \tag{26}$$

$$b_{pk+1}^{(pk+t+1)} = b_{pk+1}^{(pk+t)} - Reg^{(0)}\, b_{pk+t+1}^{(pk+t+1)}, \tag{27}$$

$$b_{pk+2}^{(pk+t+1)} = b_{pk+2}^{(pk+t)} - Reg^{(1)}\, b_{pk+t+1}^{(pk+t+1)}, \tag{28}$$

. . . . .

$$b_{pk+t}^{(pk+t+1)} = b_{pk+t}^{(pk+t)} - Reg^{(t-1)}\, b_{pk+t+1}^{(pk+t+1)} \tag{29}$$

for $pk + t + 2 \le j \le n$,

a main controller J that, if n is a multiple of k, instructs the pivot choosing section, the preprocessing sections $A_1, \ldots, A_k$, the updating section B', and the postprocessing sections $C_1, \ldots, C_{k-1}$ to repeat their above operations for $p = 0, 1, \ldots, n/k - 1$, and, if n is not a multiple of k, instructs the pivot choosing section, the preprocessing sections $A_1, \ldots, A_k$, the updating section B', and the postprocessing sections $C_1, \ldots, C_{k-1}$ to repeat their above operations for $p = 0, 1, \ldots [n/k] - 1$, and instructs the pivot choosing section, the preprocessing sections $A_1, \ldots, A_{n-[n/k]k}$, the updating section B', and the postprocessing sections $C_1, \ldots,$
$C_{n-[n/k]k}$ to execute their above operations for $p = [k/n]$.

According to another aspect of the present invention there is provided a system of nodes $\alpha_0, \ldots, \alpha_{P-1}$, each of which is connected to each other by a network and comprises:

a memory that stores blocks of k rows of each coefficient matrix $A^{(r)}$ and corresponding k components of each known vector $b^{(r)}$ and an unknown vector x expressed by (1) for a given system of linear equations (2),

a pivot choosing section that is connected to the memory, chooses a pivot in the i-th row of $A^{(i-1)}$, and interchanges the i-th column with the chosen pivotal column,

a preprocessing section A_1 that is connected to the memory and calculates (4) for $pk + 2 \le j \le n$ and (5),

k - 1 preprocessing sections A_t, where $t = 2, 3, \ldots, k$, each of which is connected to the memory, calculates (6), (7), . . . , (10) for $pk + t \le j \le n$, and calculates (12) and (13) for $pk + t + 1 \le j \le n$,

an updating section B that is connected to the memory, comprises a set of k registers and an arithmetic unit, and calculates (14), (15), . . . , (18) for $(p + 1)k + 1 \le j \le n$ retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in the register set,

a back-substitution section that is connected to the memory and obtains the unknown x by back-substitution, that is, by calculating (19) and (20),

a gateway that is connected to the memory and is a junction with the outside, and

a transmitter that is connected to the memory and transmits data between the memory and the outside through the gateway.

If the (pk + 1)th through (p + 1)k-th rows of $A^{(0)}$ and corresponding components of $b^{(0)}$ and x are assigned to the node α_u, then the pivot choosing section of the node α_u determines the pivot (3), and the preprocessing section of the node α_u calculates (4) and (5) for $pk + 2 \leq j \leq n$, and the transmitter transmits the results to the memory of every other node through the gateway, while the updating section B of the node in charge of the i-th row calculates (14) for every i such that $(p + 1)k + 1 \leq i \leq n$. This series of operations is below called parallel preprocessing A_1.

The preprocessing section A_t of the above node α_u calculates (6), (7), (8), (9), (10) for $pk + t \leq j \leq n$, and, immediately after the pivot choosing section of α_u determines the pivot (11), calculates (12) and (13) for $pk + t + 1 \leq j \leq n$, and the transmitter transmits the results to the memory of every other node through the gateway, while the updating section B of the node in charge of the i-th row calculates

$$Reg_i^{(t-1)} = a_{ipk+t}^{(pk)} - \sum_{m=1}^{k-1} Reg_i^{(m-1)} a_{pk+mpk+t}^{(pk+m)} \qquad (30)$$

for every i such that $(p + 1)k + 1 \leq i \leq n$. This series of operations is below called parallel preprocessing A_t, where $2 \leq t \leq k$.

The updating section B of each node in charge of the i-th row such that $(p + 1)k + 1 \leq i \leq n$ also calculates (14) through (18) retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in the register set. These operations are below called parallel updating B.

According to a further aspect of the present invention there is provided a main controller G_p that is connected to the system of nodes by the network, distributes and assigns the rows of the coefficient matrix $A^{(0)}$ and the components of $b^{(0)}$ and x to the nodes in such a manner as each block of consecutive k rows and corresponding 2k components is transmitted to the memory of one node in the cyclic order of $\alpha_0, \ldots, \alpha_{p-1}, \alpha_0, \alpha_1, \ldots$, and, if n is a multiple of k, instructs each node to execute parallel preprocessing A_1 through A_k and parallel updating B for $p = 0, 1, \ldots, n/k - 1$, and, if n is not a multiple of k, instructs each node to execute parallel preprocessing A_1 through A_k and parallel updating B for $p = 0, 1, \ldots, [n/k] - 1$ and to execute parallel preprocessing A_1 through $A_{n-[n/k]k}$ for $p = [n/k]$, and instructs the nodes to obtain unknown vector by means of back-substitution.

According to another aspect of the present invention there is provided a system of nodes $\alpha_0, \ldots, \alpha_{p-1}$, each of which is connected to each other by a network and comprises:

a memory that stores blocks of k rows of each coefficient matrix $A^{(r)}$ and corresponding k components of each known vector $b^{(r)}$ and an unknown vector x expressed by (1) for a given system of linear equations (2),

a pivot choosing section that is connected to the memory, chooses a pivot in the i-th row of $A^{(i-1)}$, and interchanges the i-th column with the chosen pivotal column,

a preprocessing section A_1 that is connected to the memory and calculates (4) for $pk + 2 \leq j \leq n$ and (5),

k - 1 preprocessing sections A_t, where $t = 2, 3, \ldots, k$, each of which is connected to the memory, calculates (6), (7), . . . , (10) for $pk + t \leq j \leq n$, and calculates (12) and (13) for $pk + t + 1 \leq j \leq n$,

an updating section B' that is connected to the memory, comprises a set of k registers and an

arithmetic unit, and calculates (14), (15), . . . , (18) for $(p + 1)k + 1 \leq j \leq n$ retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in the register set,

k - 1 postprocessing sections C_t, where $t = 1, 2, \ldots, k - 1$, each of which is connected to the memory and calculates (21), (22), . . . , (29) for $pk + 2 + 2 \leq j \leq n$,

a gateway that is connected to the memory and is a junction with the outside, and

a transmitter that is connected to the memory and transmits data between the memory and the outside through the gateway.

If the $(pk + 1)$th through $(p + 1)k$-th rows of $A^{(0)}$ and corresponding components of $b^{(0)}$ and x are assigned to the node α_u, then the pivot choosing section of α_u determines the pivot (3), and the preprocessing section of α_u calculates (4) and (5) for $pk + 2 \leq j \leq n$, and the transmitter transmits the results to the memory of every other node through the gateway, while the updating section B of the element processor in charge of the i-th row calculates (14) for every i such that $(p + 1)k + 1 \leq i \leq n$. This series of operations is below called parallel preprocessing A_1.

The preprocessing section A_t of the node α_u calculates (6), (7), (8), (9), (10) for $pk + t \leq j \leq n$, and, immediately after the pivot choosing section 2 of α_u determines the pivot (11), calculates (12) and (13) for $pk + t + 1 \leq j \leq n$, and the transmitter transmits the results to the memory of every other node through the gateway, while the updating section B' of the node in charge of the i-th row calculates (30) for every i such that $(p + 1)k + 1 \leq i \leq n$. This series of operations is below called parallel preprocessing A_t, where $2 \leq t \leq k$.

The updating section B' of each node in charge of the i-th row such that $1 \leq i \leq pk$ or $(p + 1)k + 1 \leq i \leq n$ if n is a multiple of k or $p < [n/k]$ and $1 \leq i \leq [n/k]k$ otherwise also calculates (14) through (18) for $(p + 1)k + 1 \leq j \leq n$ if n is a multiple of k or $p < [n/k]$ and for $[n/k]k + 1 \leq j \leq n$ otherwise, retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in the register set. These operations are below called parallel updating B'.

The postprocessing section C_t of the above node α_u calculate (21), (22), . . . , (29) for $pk + t + 2 \leq j \leq n$ for $t = 1, 2, \ldots, k - 1$ if n is a multiple of k or $p < [n/k]$ and for $t = 1, 2, \ldots, n - [n/k]k$ otherwise. This series of operations is below called post-elimination C.

According to a further aspect of the present invention there is provided a main controller J_p that is connected to the system of nodes by the network, distributes the rows of the coefficient matrix $A^{(0)}$ and the components of $b^{(0)}$ and x to the coefficient matrix $A^{(0)}$ and the components of $b^{(0)}$ and x to the nodes in such a manner as each block of consecutive k rows and corresponding 2k components is transmitted to the memory of one node in the cyclic order of $\alpha_0, \ldots, \alpha_{P-1}, \alpha_0, \alpha_1, \ldots$, and, if n is a multiple of k, instructs each node to execute parallel preprocessing A_1 through A_k, parallel updating B' and post-elimination C for $p = 0, \ldots, n/k - 1$, and, if n is not a multiple of k, instructs each node to execute parallel preprocessing A_1 through A_k, parallel updating B' and post-elimination C for $p = 0, 1, \ldots, [n/k] - 1$ and to execute parallel preprocessing A_1 through $A_{n-[n/k]k}$, parallel updating B', and post-elimination C for $p = [n/k]$.

According to another aspect of the present invention there is provided an element processor comprising:

a pivot choosing section that, for coefficient matrices $A^{(r)}$, known vectors $b^{(r)}$ and an unknown vector x expressed by (1) for a given system of linear equations (2), chooses a pivot in the i-th row of $A^{(i-1)}$ and interchanges the i-th column with the chosen pivotal column,

a preprocessing section A_1 that is connected to the pivot choosing section and calculates (4) for $pk + 2 \leq j \leq n$ and (5),

k - 1 preprocessing sections A_t, where $t = 2, 3, \ldots, k$, each of which is connected to the pivot choosing section, calculates (6), (7), . . . , (10) for $pk + t \leq j \leq n$, and calculates (12) and (13) for $pk + t + 1 \leq j \leq n$,

an updating section B which is connected to the pivot choosing section, comprises a set of k registers and an arithmetic unit, and calculates (14), (15), . . . , (18) for $(p + 1)k + 1 \leq j \leq n$ retaining the values of

$$Reg_1^{(0)}, \ldots, Reg_1^{(k)}$$

in the register set,

a back-substitution section that is connected to the pivot choosing section and obtains the unknown x by back-substitution, that is, by calculating (19) and (20), and

a gateway that is connected to the pivot choosing section and is a junction with the outside.

According to a further aspect of the present invention there is provided a system of clusters, $CL_0, \ldots, CL_{P-1}$, each of which is connected to each other by a network and comprises:

above element processors $PE_1, \ldots,$

$$PE_{p_c},$$

a memory that stores blocks of k rows of each coefficient matrix $A^{(r)}$ and corresponding k components of each known vector $b^{(r)}$ and the unknown vector x,

a C gateway that is a junction with the outside, and

a transmitter that transmits data between the memory and the outside through the C gateway.

If the (pk + 1)th through (p + 1)k-th rows of $A^{(0)}$ and corresponding components of $b^{(0)}$ and x are assigned to the cluster CL_u, then the pivot choosing section, the updating section and the back-substitution section of each element processor of CL_u take charge of part of the k rows and 2k components row by row, while the preprocessing section A_t of each element processor of CL_u takes charge of elements of the (pk + t)th row of $A^{(r)}$ and the (pk + t)th component of $b^{(r)}$ one by one.

Specifically, the pivot choosing section of the element processor PE_1 of CL_u determines the transposed pivot (3) of the (pk + 1)th row, and the preprocessing sections A_1 of element processors of CL_u simultaneously calculate (4) and (5) for $pk + 2 \leq j \leq n$ and (5) with each A_1 calculating for elements and components in its charge, and the transmitter transmits the results to the memory of every other cluster through the C gateway, while the updating section B of the element processor in charge of the i-th row calculates (14) for every i such that $(p + 1)k + 1 \leq i \leq n$. This series of operations is below called parallel preprocessing CLA_1.

The preprocessing sections A_t of the above cluster CL_u simultaneously calculate (6), (7), (8), (9), (10) for $pk + t \leq j \leq n$ with each A_t calculating for elements and components in its charge, immediately after the pivot choosing section of PE_t of CL_u determines the pivot (11), simultaneously calculate (12) and (13) for $pk + t + 1 \leq j \leq n$, and the transmitter transmits the results to the memory of every other cluster through the C gateway, while the updating section B of the element processor in charge of the i-th row calculates (30) for every i such that $(p + 1)k + 1 \leq i \leq n$. This series of operations is below called parallel preprocessing CLA_t, where $2 \leq t \leq k$.

The updating sections B of each element processor in charge of the i-th row such that $(p + 1)k + 1 \leq i \leq n$ calculate (14) through (18) for $(p + 1)k + 1 \leq j \leq n$ retaining the values of

$$Reg_1^{(0)}, \ldots, Reg_1^{(k)}$$

in the register set. These operation are below called parallel updating B_c.

According to a further aspect of the present invention there is provided a main controller G_{pc} that is connected to the above system, distributes and assigns the rows of the coefficient matrix $A^{(0)}$ and the components of $b^{(0)}$ and x to the clusters in such a manner as each block of consecutive k rows and corresponding 2k components is transmitted to the memory of one cluster in the cyclic order of $CL_0, \ldots, CL_{P-1}, CL_0, CL_1, \ldots,$ and, if n is a multiple of k, instructs each cluster to execute parallel preprocessing CLA_1 through CLA_k and parallel updating B_c for $p = 0, 1, \ldots, n/k - 2$ and to execute CLA_1 through CLA_k for $p = n/k - 1$, and, if n is not a multiple of k, instructs each cluster to execute CLA_1 through CLA_k and B_c for $p = 0, 1, \ldots, [n/k] - 1$ and to execute CLA_1 through $CLA_{n-[n/k]k}$ for $p = [n/k]$, and instructs each cluster to obtain the unknown vector x by means of the back-substitution sections of its element processors and its transmitter.

According to another aspect of the present invention there is provided an element processor compris-

ing:

a pivot choosing section that, for coefficient matrices $A^{(r)}$, known vectors $b^{(r)}$ and an unknown vector x expressed by (1) for a given system of linear equations (2), chooses a pivot in the i-th row of $A^{(i-1)}$ and interchanges the i-th column with the chosen pivotal column,

a preprocessing section A_1 that is connected to the pivot choosing section and calculates (4) for $pk + 2 \leq j \leq n$ and (5),

k - 1 preprocessing sections A_t, where $t = 2, 3, \ldots, k$, each of which is connected to the pivot choosing section, calculates (6), (7), . . . , (10) for $pk + t \leq j \leq n$, and calculates (12) and (13) for $pk + t + 1 \leq j \leq n$,

an updating section B' which is connected to the pivot choosing section, comprises a set of k registers and an arithmetic unit, and calculates (14), (15), . . . , (18) for $(p + 1)k + 1 \leq j \leq n$ retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in the register set,

k - 1 postprocessing sections C_t, where $t = 1, 2, \ldots, k - 1$, each of which is connected to the pivot choosing section and calculates (21), (22). . . , (29) for $pk + t + 2 \leq j \leq n$, and

a gateway that is connected to the pivot choosing section and is a junction with the outside.

According to a further aspect of the present invention there is provided a system of clusters, $CL_0, \ldots, CL_{P-1}$, each of which is connected to each other by a network and comprises:

above element processors $PE_1, \ldots,$

$$PE_{P_C},$$

a memory that stores the coefficient matrices $A^{(r)}$, the known vectors $b^{(r)}$ and the unknown vector x,

a C gateway that is a junction with the outside, and

a transmitter that transmits data between the memory and the outside through the C gateway.

If the (pk + 1)th through (p + 1)k-th rows of $A^{(0)}$ and corresponding components of $b^{(0)}$ and x are assigned to the cluster CL_u, then the pivot choosing section and the updating section B' of each element processor of CL_u take charge of part of the k rows and 2k components row by row, while the preprocessing section A_t and postprocessing section C_t of each element processor of CL_u take charge of elements of the (pk + t)th row of $A^{(r)}$ and the (pk + t)th component of $b^{(r)}$ one by one.

Specifically, the pivot choosing section of the element processor PE_1 of CL_u determines the transposed pivot (3) of the (pk + 1)th row, and the preprocessing sections A_1 of element processors of CL_u simultaneously calculate (4) and (5) for $pk + 2 \leq j \leq n$ with each A_1 calculating for elements and components in its charge, and the transmitter transmits the results to the memory of every other cluster through the C gateway, while the updating section B' of the element processor in charge of the i-th row calculates (14) for every i such that $(p + 1)k + 1 \leq i \leq n$. This series of operations is below called parallel preprocessing CLA_1.

The preprocessing sections A_t of element processors of the above cluster CL_u simultaneously calculate (6), (7), (8), (9), (10) for $pk + t \leq j \leq n$ with each A_t calculating for elements and components in its charge and, immediately after the pivot choosing section of PE_t of CL_u determines the pivot (11), simultaneously calculate (12) and (13) for $pk + t + 1 \leq j \leq n$, and the transmitter transmits the results to the memory of every other cluster through the C gateway, while the updating section B' of the element processor in charge of the i-th row calculates (30) for every i such that $(p + 1)k + 1 \leq i \leq n$. This series of operations is below called parallel preprocessing CLA_t, where $2 \leq t \leq k$.

The updating section B' of each element processor in charge of the i-th row such that $1 \leq i \leq pk$ or $(p + 1)k + 1 \leq i \leq n$ if n is a multiple of k or $p < [n/k]$ and $1 \leq i \leq [n/k]k$ otherwise also calculates (14) through (18) for $(p + 1)k + 1 \leq j \leq n$ if n is a multiple of k or $p < [n/k]$ and for $[n/k]k + 1 \leq j < n$ otherwise, retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in the register set. These operations are below called parallel updating B'_c.

The postprocessing sections C_t of element processors of the above CL_u simultaneously calculate (21), (22), . . . , (29) for j such that $pk + t + 2 \leq j \leq n$ for $t = 1, 2, \ldots, k - 1$ if n is a multiple of k or $p < [n/k]$ and for $t = 1, 2, \ldots, n - [n/k]k$ otherwise. This series of operations is below called postelimination C_c.

According to a further aspect of the present invention there is provided a main controller J_{pc} that is connected to the above system, distributes and assigns the rows of the coefficient matrix $A^{(0)}$ and the components of $b^{(0)}$ and x to the clusters in such a manner as each block of consecutive k rows and corresponding 2k components is transmitted to the memory of one cluster in the cyclic order of $CL_0, \ldots, CL_{P-1}, CL_0, CL_1, \ldots$, and, if n is a multiple of k, instructs each cluster to execute parallel preprocessing CLA_1 through CLA_k, parallel updating B'_c and parallel postelimination C_c for $p = 0, 1, \ldots, n/k - 1$, and if n is not a multiple of k, instructs each cluster to execute parallel preprocessing CLA_1 through CLA_k, parallel updating B'_c , and post-elimination C_c for $p = 0, 1, \ldots, [n/k] - 1$ and to execute parallel preprocessing CLA_1 through $CLA_{n-[n/k]k}$, parallel updating B'_c, and postelimination C_c for $p = [n/k]$.

According to another aspect of the present invention, there is provided a parallel elimination method for solving the system of linear equations (2) in a parallel computer comprising C clusters $CL_1, \ldots, CL_C$ connected by a network. Each of the clusters comprises P_c element processors and a shared memory that stores part of the reduced matrices $A^{(r)}$ and the known vectors $b^{(r)}$ and the unknown vector x. The method comprises:

a data distribution means that distributes the rows of the coefficient matrix $A^{(0)}$ and the components of $b^{(0)}$ and x to the shared memory of the clusters in such a manner as each block of consecutive k rows and corresponding 2k components is transmitted to the shared memory in the cyclic order of $CL_1, \ldots, CL_C, CL_1, CL_2, \ldots$, and assigns those distributed to the cluster's shared memory to its element processors row by row,

a pivot choosing means that chooses a pivot in a row assigned to each element processor,

an elementary pre-elimination means that, after the pivot choosing means chooses the pivot

$$a_{kP_c+1\,kP_c+1}^{(kP_c)}, \tag{31}$$

calculates

$$a_{kP_c+1\,j}^{(kP_c+1)} = a_{kP_c+1\,j}^{(kP_c)} / a_{kP_c+1\,kP_c+1}^{(kP_c)}, \tag{32}$$

$$b_{kP_c+1}^{(kP_c+1)} = b_{kP_c+1}^{(kP_c)} / a_{kP_c+1\,kP_c+1}^{(kP_c)} \tag{33}$$

in the element processor in charge of the $(kP_c + 1)$th row, transmits the results to the shared memory of every other cluster to which the element processor in charge of an i-throw such that $kP_c + 1 \leq i \leq n$ belongs, and, for $l = 2, \ldots, P_c$, calculates

$$t_l^{(l-1)} = a_{i\,kP_c+l}^{(kP_c)} - a_{i\,kP_c+1}^{(kP_c)} a_{kP_c+1\,kP_c+l}^{(kP_c+1)} - \sum_{m=2}^{l-1} t_l^{(m-1)} a_{kP_c+m\,kP_c+l}^{(kP_c+m)} \tag{34}$$

for $kP_c + l \leq i \leq n$ in the element processor in charge of the i-th row, calculates

$$a_{kP_c+1j}^{(kP_c+l-1)} = a_{kP_c+1j}^{(kP_c)} - a_{kP_c+1kP_c+1}^{(kP_c)} a_{kP_c+1j}^{(kP_c+1)} - \sum_{m=2}^{l-1} t_{kP_c+1}^{(m-1)} a_{kP_c+mj}^{(kP_c+m)}, \quad (35)$$

$$b_{kP_c+1}^{(kP_c+l-1)} = b_{kP_c+1}^{(kP_c)} - a_{kP_c+1kP_c+1}^{(kP_c)} b_{kP_c+1}^{(kP_c+1)} - \sum_{m=2}^{l-1} t_{kP_c+1}^{(m-1)} b_{kP_c+m}^{(kP_c+m)} \quad (36)$$

in the element processor in charge of the $(kP_c + l)$th row, and, after the pivot choosing means determines the pivot

$$a_{kP_c+1kP_c+1}^{(kP_c+l-1)}, \quad (37)$$

calculates

$$a_{kP_c+1j}^{(kP_c+l)} = a_{kP_c+1j}^{(kP_c+l-1)} / a_{kP_c+1kP_c+1}^{(kP_c+l-1)}, \quad (38)$$

$$b_{kP_c+1}^{(kP_c+l)} = b_{kP_c+1}^{(kP_c+l-1)} / a_{kP_c+1kP_c+1}^{(kP_c+l-1)} \quad (39)$$

in the element processor in charge of the $(kP_c + l)$th row, transmits the results (38) and (39) to the shared memory of every other cluster to which the element processor in charge of an i-th row such that $kP_c + l + 1 \leq i \leq n$ belongs,

a multi-pivot elimination means that calculates

$$a_{ij}^{((k+1)P_c)} = a_{ij}^{(kP_c)} - a_{ikP_c+1}^{(kP_c)} a_{kP_c+1j}^{(kP_c+1)} - \sum_{m=2}^{P_c} t_i^{(m-1)} a_{kP_c+mj}^{(kP_c+m)}, \quad (40)$$

$$b_i^{((k+1)P_c)} = b_i^{(kP_c)} - a_{ikP_c+1}^{(kP_c)} b_{kP_c+1}^{(kP_c+1)} - \sum_{m=2}^{P_c} t_i^{(m-1)} b_{kP_c+m}^{(kP_c+m)} \quad (41)$$

in each element processor in charge of the i-th row such that $(k + 1)P_c + 1 \leq i \leq n$,

a means for testing if the operation of the multi-pivot elimination means was repeated $[n/P_c]$ times, and

a remainder elimination means that executes the above elementary pre-elimination means for the $(-[n/P_c]P_c + 1)$th row through the n-th row, if the above testing means judges that the operation of the multi-pivot elimination means was executed $[n/P_c]$ times, and n is not a multiple of P_c.

According to a further aspect of the present invention, there is provided a parallel computation method comprising:

an elementary back-substitution means that calculates

$$x_i = b_i^{(n)} \tag{42}$$

in the element processor in charge of the i-th row after the elimination process of the above parallel elimination method,

an elementary back-transmission means that transmits x_i to the shared memory of every cluster to which the element processor in charge of an h-th row such that $1 \leq h \leq i - 1$ belongs,

an elementary back-calculation means that calculates

$$b_h^{(n+h-i+1)} = b_h^{(n+h-i)} - a_{hi}^{(h)} x_i, \tag{43}$$

for $1 \leq h \leq i - 1$ in the element processor in charge of the h-th row, and

a means for testing if the operation of the elementary back-substitution means was repeated from $i = n$ to $i = 1$.

The solution of the system of linear equation (1) is thus obtained by the elementary back-substitution as

$$x_n = b_n^{(n)}, \ldots, x_1 = b_1^{(n)} \tag{44}$$

in this order.

According to another aspect of the present invention, there is provided a parallel elimination method for solving the system of linear equations (2) in a parallel computer comprising C clusters $CL_1, \ldots, CL_C$ connected by a network. Each of the clusters comprises P_c element processors and a shared memory that stores part of the reduced matrices $A^{(r)}$ and the known vectors $b^{(r)}$ and the unknown vector x. The method comprises:

a data distribution means that distributes the rows of the coefficient matrix $A^{(0)}$ and the components of $b^{(0)}$ and x to the clusters in such a manner as each block of consecutive k rows and corresponding 2k components is transmitted to the shared memory in the cyclic order of $CL_1, \ldots, CL_C, CL_1, CL_2, \ldots$, and assigns those distributed to the cluster's shared memory to its element processors row by row,

a pivot choosing means that chooses a pivot in a row assigned to each element processor,

an elementary pre-elimination means that, after the pivot choosing means chooses the pivot (31), calculates (32) and (33) in the element processor in charge of the $(P_c k + 1)$th row, transmits the results to the shared memory of every other cluster to which the element processor in charge of an i-th row such that $kP_c + 2 \leq i \leq n$ belongs, and, for $l = 2, \ldots, P_c$, calculates (34) for $kP_c + l \leq i \leq n$ in the element processor in charge of the i-th row, calculates (35) and (36) in the element processor in charge of the $(kP_c + l)$th row, and, after the pivot choosing means chooses the pivot (37), calculates (38) and (39) in the element processor in charge of the $(kP_c + l)$th row, and transmits the results (38) and (39) to the shared memory of every other cluster to which an element processor in charge of the i-th row such that $kP_c + l + 1 \leq i \leq n$ belongs, calculates,

a multi-pivot elimination means that calculates (43) and (44) in each element processor in charge of the i-throw such that $(k + 1)P_c + 1 \leq i \leq n$,

an elementary post-elimination means that calculates

$$a_{ij}^{(r+1)} = a_{ij}^{(r)} - a_{il+1}^{(r)} a_{l+1j}^{(r+1)}, \tag{45}$$

$$b_i^{(r+1)} = b_i^{(r)} - a_{il+1}^{(r)} b_{l+1}^{(r+1)} \tag{46}$$

in the element processor in charge of the i-th row,

a post-elimination processing means that calculates (45) and (46) for $l = -w + q + 1$ for $w = 1, \ldots, q$ and $q = 1, \ldots, P_c - 1$ for $kP_c + 1 \leq i \leq kP_c + q$ in the element processor in charge of the i-th row,

a means for testing if the operation of the post-elimination means was executed $[n/P_c]$ times, and

a remainder elimination means that executes the above elementary pre-elimination means for the $(-[n/P_c]P_c + 1)$th through the n-th rows and executes the above multi-pivot elimination means and the post-elimination means, if the above testing means judges that the operation of the post-elimination means was executed $[n/P_c]$ times.

According to a further aspect of the present invention, there is provided

a search means whereby an above element processor searches for a nonzero element in the order of increasing column numbers from that diagonal element in the same row, if a diagonal element of a coefficient matrix is 0,

a column number broadcasting means that notifies other element processors of the column number of a nonzero element found by the above search means,

an element interchange means whereby each element processor interchanges the two elements which are in its charge and have the same column numbers as the above diagonal zero element and the found nonzero element, and

a component interchange means whereby two element processors interchange the two components of the unknown vector which are in their charge and have the same component indices as the column numbers of the above diagonal zero element and the found nonzero element.

According to a further aspect of the present invention, there is provided

a search means whereby an above element processor searches for an element with the greatest absolute value in the order of increasing column numbers from a diagonal element in the same row,

a column number broadcasting means that notifies other element processors of the column number of an element found by the above search means,

an element interchange means whereby each element processor interchanges the two elements which are in its charge and have the same column number as the above diagonal element and the found element, and

a component interchange means whereby two element processors interchange the two components of the unknown vector which are in their charge and have the same component indices as the column numbers of the above diagonal element and the found component.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other objects and features of the present invention will become clear from the following description taken in conjunction with the preferred embodiments thereof with reference to the accompanying drawings throughout which like parts are designated by like reference numerals, and in which:

Fig. 1 is a block diagram of a linear calculating equipment according to the first embodiment of the present invention.

Fig. 2 is a flow chart of a control algorithm to be performed in the first embodiment.

Fig. 3 is a block diagram of a linear calculating equipment according to the second embodiment of the present invention.

Fig. 4 is a flow chart of the control algorithm to be performed in the second embodiment.

Fig. 5 is a block diagram of a parallel linear calculating equipment according to the third embodiment of the present invention.

Fig. 6 is a block diagram of a node shown in Fig. 5.

Fig. 7 is a flow chart of the control algorithm to be performed in the third embodiment.

Fig. 8 is a block diagram of a parallel linear calculating equipment according to the fourth embodiment of the present invention.

Fig. 9 is a block diagram of a node shown in Fig. 8.

Fig. 10 is a flow chart of the control algorithm to be performed in the fourth embodiment.

Fig. 11 is a block diagram of a parallel linear calculating equipment according to the fifth embodiment of the present invention.

Fig. 12 is a block diagram of a cluster shown in Fig. 11.

Fig. 13 is a block diagram of an element processor shown in Fig. 12.

Fig. 14 is a flow chart of the control algorithm to be performed in the fifth embodiment.

Fig. 15 is a block diagram of a parallel linear calculating equipment according to the sixth embodiment of the present invention.

Fig. 16 is a block diagram of a cluster shown in Fig. 15.

Fig. 17 is a block diagram of an element processor shown in Fig. 16.
Fig. 18 is a flow chart of the control algorithm to be performed in the sixth embodiment.
Fig. 19 is a block diagram of an element processor or processor module in a parallel computer which implements the 7th and 8th embodiments.
Fig. 20 is a block diagram of a cluster used in the 7th and 8th embodiments.
Fig. 21 is a block diagram of the parallel computation method according to the 7th embodiment.
Fig. 22 is a block diagram of the parallel computation method according to the 8th embodiment.
Fig. 23 is a diagram for showing the pivoting method according to the 7th and 8th embodiments.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The preferred embodiments according to the present invention will be described below with reference to the attached drawings.

Fig. 1 is a block diagram of linear calculating equipment in the first embodiment of the present invention. In Fig. 1, 1 is a memory; 2 is a pivoting section connected to the memory 1; 3, 4, 5 are preprocessing sections A_1, A_t, A_k respectively, each connected to the memory 1; 6 is an updating section B connected to the memory 1; 7 is a back-substitution section connected to the memory 1; 8 is a main controller G; 101 is a register set composed of k registers; 102 is an arithmetic unit.

Following is a description of the operation of each component of the first embodiment.

The memory 1 is ordinary semiconductor memory and stores reduced coefficient matrices $A^{(r)}$ with zeroes generated from the first to the r-th column and corresponding known vectors $b^{(r)}$ and an unknown vector x expressed by (1) for a given system of linear equations (2).

The pivoting section is connected to the memory 1, chooses a pivot in the i-th row following the instruction of the main controller G 8 when the first (i - 1) columns are already reduced, and interchanges the i-th column with the chosen pivotal column and the i-th component with the corresponding component of x. The choice of the pivot is based on a method called partial pivoting whereby an element with the largest absolute value in the i-th row is chosen as the pivot. The interchange can be direct data transfer or transposition of column numbers and component indices.

Immediately after the pivoting section 2 determines the transposed pivot (3), the preprocessing section A_1 3 calculates (4) for $pk + 2 \leq j \leq n$ and (5) following the instruction of the main controller G. Each preprocessing sections A_t 4, where $t = 2, 3, \ldots, k$, is connected to the memory 1, calculates (6), (7), (8), (9), (10) for $pk + t \leq j \leq n$, and, immediately after the pivoting section determines the transposed pivot (11), calculates (12) and (13) for $pk + t + 1 \leq j \leq n$ following the instruction of the main controller G 8.

The updating section B 6 is connected to the memory 1, comprises a register set 101 of k registers and an arithmetic unit 102, and calculates (14), (15), (16), (17), (18) for $(p + 1)k + 1 \leq i, j \leq n$ in the arithmetic unit 102, retaining each value of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in the corresponding register of the register set 101 following the instruction of the main controller G 8. (14), (15), (16) are preliminary formulas, and (17) and (18) are formulas that determine updated components.

The back-substitution section 7 is connected to the memory 1 and obtains the value of the unknown vector x by calculating (19) and (20) for $1 \leq h \leq i - 1$ for $i = n, n - 1, \ldots, 1$ in this order of i.

The operation of the main controller G 8 is described below with reference to Fig. 2, which shows a flow chart of its control algorithm.

The first step tests if n is a multiple of k. If it is, then the next step initializes p as $p = 0$ and enters the loop of the left side. The t-th step within this loop where, $t = 1, \ldots, k$, instructs the pivoting section 2 and the preprocessing section A_t 4 to execute their operations for the (pk + t)th row of the current reduced matrix $A^{(pk+t-1)}$. The next step tests if $p = n/k - 1$. If it is, then the next step escapes the loop. If $p < n/k - 1$, then the next step instructs the updating section B 6 to execute its operation. The next step increments p by 1 and returns to the operations of the pivoting section 2 and the preprocessing section A_1 3.

If n is not a multiple of k, then the next step initializes p as $p = 0$ and enters the loop of the right side. Within this loop, the operations are the same except the fact that the condition for escaping the loop is $p = [n/k]$, and the position of the testing for escape is immediately after the operation of $A_{n-[n/k]k}$.

After escaping one of the loops the final step instructs the back-substitution section 7 to execute its operation and terminates the whole operation to obtain the unknown vector x.

Fig. 3 is a block diagram of linear calculating equipment in the second embodiment of the present invention. In Fig. 3, 1 is a memory, 2 is a pivoting section connected to the memory 1; 3, 4, 5 are preprocessing sections A_1, A_t, A_k respectively, each connected to the memory 1; 9 is an updating section B' connected to the memory 1; 10, 11, 12 are postprocessing sections C_1, C_t, C_{k-1} respectively, each connected to the memory 1; 13 is a main controller J; 103 is a register set composed of k registers; 104 is an arithmetic unit for, 101 is an arithmetic unit.

Following is a description of the operation of each component, which is different from one in the first embodiment.

The updating section B' 9 is connected to the memory 1 and calculates (14), (15), . . . , (18) for $1 \leq i \leq pk$, $(p + 1)k + 1 \leq i \leq n$, $(p + 1)k + 1 \leq j \leq n$ if n is a multiple of k or $p < [n/k]$ and for $1 \leq i \leq [n/k]k$, $[n/k]k + 1 \leq j \leq n$ otherwise in the arithmetic unit 104, retaining each value of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in the corresponding register of the register set 103.

The k - 1 postprocessing sections C_t 11, where $t = 1, 2, \ldots, k - 1$, are connected to the memory 1 and calculate (21), (22), ... , (29) for $pk + t + 2 \leq j \leq n$.

The operation of the main controller J 13 is described below with reference to Fig. 4, which shows a flow chart of its control algorithm.

The first step tests if n is a multiple of k. If it is, then the next step initializes p as $p = 0$ and enters the left side loop. The t-th step within this loop, where $t = 1, \ldots, k$, instructs the pivoting section 2 and the preprocessing section A_t 4 to execute their operations for the (pk + t)th row of the current reduced matrix $A^{(pk+t-1)}$.The next step instructs the updating section B' 9 to execute its operation. The following k - 1 steps instruct the postprocessing sections C_1 10 through C_{k-1} 12 to execute their operations in this order. The next step tests if $p = n/k - 1$. If it is, then the next step escapes the loop and terminates operation. If $p < n/k - 1$, then the next step increments p by 1 and returns to the operation of the pivoting section 2.

If n is not a multiple of k, then the next step initializes p as $p = 0$ and enters the right side loop. Within this loop, the first $n - [n/k]k + 1$ steps are the same as those in the loop of the left side. After instructing the preprocessing section $A_{n-[n/k]k}$ 4 to execute its operation, the step tests if $p = [n/k]$. If it is not, then the following steps order the operations of the pivoting section 2 and the preprocessing section $A_{n-[n/k]k+1}$ 4 through the operations of the pivoting section 2 and the preprocessing section A_k 5 followed by the operation of the updating section B'9 and then the operations of the postprocessing sections C_1 10 through C_{k-1} 12. Then the step increments p by 1 and returns to the operation of the pivoting section 2. If $p = [n/k]$, then the following steps instruct the updating section B' 9 to execute its operation, instruct the postprocessing sections C_1 10 through $C_{n-[n/k]k}$ 11 to execute their operations, and terminates the whole process to obtain the unknown vector.

Fig. 5 is a block diagram of parallel linear calculating equipment in the third embodiment of the present invention. In Fig. 5, 21 is a network; 22, 23, 24 are nodes α_0, α_u, α_{P-1} mutually connected by the network 21; 25 is a main controller G_p connected to each node. Fig. 6 is a block diagram of a node in Fig. 5. In Fig. 6, 1 is a memory; 2 is a pivoting section connected to the memory 1; 3, 4, 5 are preprocessing sections A_1, A_t, A_k respectively, each connected to the memory 1; 6 is an updating section B connected to the memory 1; 7 is a back-substitution section connected to the memory 1; 26 is a gateway that is a junction with the outside ; 27 is a transmitter that transmits data between the memory 1 and the outside through the gateway 26; 101 is a register set composed of k registers; 102 is an arithmetic unit.

Following is a description of the operation of each component of the third embodiment.

If the (pk + 1)th through (p + 1)k-th rows of $A^{(0)}$ and corresponding components of $b^{(0)}$ and x are assigned to the node α_u 23, then the pivoting section 2 of the node α_u 23 determines the pivot (3), and the preprocessing section of the node α_u 23 calculates (4) and (5) for $pk + 2 \leq j \leq n$, and the transmitter 27 transmits the results to the memory 1 of every other node through the gateway 26, while the updating section B 6 of the element processor in charge of the i-th row calculates (14) for every i such that $(p + 1)k + 1 \leq i \leq n$. This series of operations is below called parallel preprocessing A_1.

The preprocessing section A_t 4 of the node α_u 23 calculates (6), (7), (8), (9), (10) for $pk + t \leq j \leq n$, and, immediately after the pivoting section 2 of α_u 23 determines the pivot (11), calculates (12) and (13) for $pk + t + 1 \leq j \leq n$, and the transmitter 27 transmits the results to the memory 1 of every other node through the gateway 26, while the updating section B 6 of the element processor in charge of the i-th row calculates (30) for every i such that $(p + 1)k + 1 \leq i \leq n$. This series of parallel operations is below called parallel

preprocessing A_t, where $2 \leq t \leq k$.

The updating section B 6 of each node in charge of the i-th row such that $(p + 1)k + 1 \leq i \leq n$ also calculates (14) through (18) for $(p + 1)k + 1 \leq j \leq n$ retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in the register set. These operations are below called parallel updating B.

The back-substitution sections 7 of nodes α_u 23 calculate (19) and (20) using necessary data transmitted by the transmitters 27 of other nodes. These operations are called back-substitution.

The operation of the main controller G_p 25 is described below with reference to Fig. 7, which shows a flow chart of its control algorithm at the level of above definition.

The first step distributes and assigns the rows of the coefficient matrix $A^{(0)}$ and the components of $b^{(0)}$ and x to the nodes α_0 22,. . . , α_u 23, . . . , α_{P-1} 24 in such a manner as each block of k rows and corresponding 2k components (n - [n/k]k rows and 2(n - [n/k]k) components in the final distribution) are transmitted to the memory 1 of one node at a time in the cyclic order of $\alpha_0, \ldots, \alpha_{P-1}, \alpha_0, \alpha_1, \ldots$.

The next step tests if n is a multiple of k. If it is, then the next step initializes p as $p = 0$ and enters the loop of the left side. The t-th step within this loop orders the execution of the parallel preprocessing A_t for the (pk + t)th row of the current reduced matrix $A^{(pk+t-1)}$.The next step tests if $p = n/k - 1$. If it is, then the next step escapes the loop. If $p < n/k - 1$, then the next step orders the execution of the parallel updating B. The next step increments p by 1 and returns to the execution of the parallel preprocessing A_1.

If n is not a multiple of k, then the next step initializes p as $p = 0$ and enters the loop of the right side. Within this loop, the operations are the same except the fact that the condition for escaping the loop is $p = [n/k]$, and the position of the testing for escape is between the parallel preprocessing $A_{n-[n/k]k}$ and $A_{n-[n/k]k+1}$.

After escaping one of the loops the final step orders the execution of back-substitution and terminates the whole operation to obtain the unknown vector x.

Fig. 8 is a block diagram of parallel linear calculating equipment in the fourth embodiment of the present invention. In Fig. 8, 31 is a network; 32, 33, 34 are nodes α_0, α_u, α_{P-1} mutually connected by the network 31; 35 is a main controller J_p connected to each node. Fig. 9 is a block diagram of a node in Fig. 8. In Fig. 9, 1 is a memory; 2 is a pivoting section connected to the memory 1; 3, 4, 5 are preprocessing sections A_1, A_t, A_k respectively, each connected to the memory 1; 9 is an updating section B' connected to the memory 1; 10, 11, 12 are postprocessing sections C_1, C_t, C_{k-1} respectively, each connected to the memory 1; 26 is a gateway that is a junction with the outside ; 27 is a transmitter that transmits data between the memory 1 and the outside through the gateway 26; 103 is a register set composed of k registers; 104 is an arithmetic unit.

Following is a description of the operation of each component of the fourth embodiment.

If the (pk + 1)th through (p + 1)k-th rows of $A^{(0)}$ and corresponding components of $b^{(0)}$ and x are assigned to the node α_u 33, then the pivoting section 2 of the node α_u 33 determines the pivot (3), and the preprocessing section of the node α_u 33 calculates (4) and (5) for $pk + 2 \leq j \leq n$, and the transmitter 27 transmits the results to the memory 1 of every other node through the gateway 26, while the updating section B 6 of the element processor in charge of the i-th row calculates (14) for every i such that $(p + 1)k + 1 \leq i \leq n$. This series of operations is below called parallel preprocessing A_1.

The preprocessing section A_t 4 of the node α_u 23 calculates (6), (7), (8), (9), (10) for $pk + t \leq j \leq n$, and, immediately after the pivoting section 2 of α_u 23 determines the pivot (11), calculates (12) and (13) for $pk + t + 1 \leq j \leq n$, and the transmitter 27 transmits the results to the memory 1 of every other node through the gateway 26, while the updating section B' 9 of the element processor in charge of the i-th row calculates (30) for every i such that $(p + 1)k + 1 \leq i \leq n$. This series of operations is below called parallel preprocessing A_t, where $2 \leq t \leq k$.

The updating section B' 9 of each node in charge of the i-th row such that $1 \leq i \leq pk$ or $(p + 1)k + 1 \leq i \leq n$ if n is a multiple of k or $p < [n/k]$ and $1 \leq i \leq [n/k]k$ otherwise also calculates (14) through (18) for $(p + 1)k + 1 \leq j \leq n$ if n is a multiple of K or $p < [n/k]$ and for $[n/k]k + 1 \leq j \leq n$ otherwise, retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in the register set. These operations are below called parallel updating B'.

The postprocessing section C_t 11 of the above node α_u 33 calculate (21), (22), . . . , (29) for $pk + t + 2 \leq j \leq n$ for $t = 1, 2, \ldots, k - 1$ if n is a multiple of k or $p < [n/k]$ and for $t = 1, 2, \ldots, n - [n/k]k$ otherwise. This series of operations is below called post-elimination C.

The operation of the main controller J_p 35 is described below with reference to Fig. 10, which shows a flow chart of its control algorithm at the level of above definition.

The first step distributes and assigns the rows of the coefficient matrix $A^{(0)}$ and the components of $b^{(0)}$ and x to the nodes α_0 32, . . . , α_u 33, . . . , α_{P-1} 34 in such a manner as each block of k rows and corresponding 2k components (n - [n/k]k rows and 2(n - [n/k]k) components in the final distribution) are transmitted to the memory 1 of one node at a time in the cyclic order of $\alpha_0, \ldots, \alpha_{P-1}, \alpha_0, \alpha_1, \ldots$.

The next step tests if n is a multiple of k. If it is, then the next step initializes p as $p = 0$ and enters the loop of the left side. The t-th step within this loop orders the execution of the parallel preprocessing A_t for the (pk + t)th row of the current reduced matrix $A^{(pk+t-1)}$.The next step orders the execution of the parallel updating B'. The next step orders the execution of the post-elimination C. The next step tests if $p = n/k - 1$. If it is, then the next step escapes the loop. If $p < n/k - 1$, then the next step increments p by 1 and returns to the execution of the parallel preprocessing A_1.

If n is not a multiple of k, then the next step initializes p as $p = 0$ and enters the loop of the right side. Within this loop, the operations are the same except the fact that the condition for escaping the loop is $p = [n/k]$, and if $p = [n/k]$, the steps skip the order for the execution of the parallel preprocessing $A_{n-[n/k]k+1}$ through A_k.

By the above processing, the unknown vector is obtained.

Fig. 11 is a block diagram of a parallel linear calculating equipment according to the fifth embodiment of the present invention. In Fig. 11, 41 is a network; 42, 43, 44 are clusters CL_0, CL_u, CL_{P-1} mutually connected by the network 41; 45 is a main controller G_{pc} connected to each cluster. Fig. 12 is a block diagram of a cluster in Fig. 11. In Fig. 12, 1 is a memory; 46 is a C gateway that is a junction with the outside; 47, 48, 49 are element processors PE_1, PE_2,

$$PE_{P_C},$$

each connected to the memory 1 ; 50 is a transmitter that transmits data between the memory 1 and the outside through the C gateway 46. Fig. 13 is a block diagram of an element processor in Fig. 12. In Fig. 13, 2 is a pivoting section; 3, 4, 5 are preprocessing sections A_1, A_t, A_k respectively, each connected to the pivoting section 2; 6 is an updating section B connected to the pivoting section 2; 7 is a back-substitution section connected to the pivoting section 2; 51 is a gateway that is a junction with the outside ; 101 is a register set composed of k registers; 102 is an arithmetic unit.

Following is a description of the operation of each component of the fifth embodiment.

If the (pk + 1)th through (p + 1)k-th rows of $A^{(0)}$ and corresponding components of $b^{(0)}$ and x are assigned to the cluster CL_u 43, then the pivoting section 2, the updating section 6 and the back-substitution section 7 of each element processor of CL_u 43 take charge of part of the k rows and 2k components row by row, while the preprocessing section A_t 4 of each element processor of CL_u 43 takes charge of elements of the (pk + t)th row of $A^{(r)}$ and the (pk + t)th component of $b^{(r)}$ one by one.

Specifically, the pivoting section 2 of the element processor PE_1 of CL_u 43 determines the transposed pivot (3) of the (pk + 1)th row, and the preprocessing sections A_1 3 of element processors of CL_u simultaneously calculate (4) and (5) for $pk + 2 \leq j \leq n$ with each A_1 3 calculating for elements and components in its charge, and the transmitter 50 transmits the results to the memory of every other cluster through the C gateway 46, while the updating section B 6 of the element processor in charge of the i-th row calculates (14) for every i such that $(p + 1)k + 1 \leq i \leq n$. This series of operations is below called parallel preprocessing CLA_1.

The preprocessing sections A_t 4 of the above cluster CL_u 43 simultaneously calculate (6), (7), (8), (9), (10) for $pk + t \leq j \leq n$ with each A_t 4 calculating for elements and components in its charge and, immediately after the pivoting section of PE_t of CL_u 43 determines the pivot (11), simultaneously calculate (12) and (13) for $pk + t + 1 \leq j \leq n$, and the transmitter 50 transmits the results to the memory 1 of every other cluster through the C gateway 46, while the updating section B 6 of the element processor in charge of the i-th row calculates (30) for every i such that $(p + 1)k + 1 \leq i \leq n$. This series of operations is below called parallel preprocessing CLA_t, where $2 \leq t \leq k$.

The updating sections B 6 of each element processor in charge of the i-th row such that $(p + 1)k + 1$

$\leq i \leq n$ calculate (14) through (18) for $(p + 1)k + 1 \leq j \leq n$ retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in the register set 101. These operations are below called parallel updating B_c.

The back-substitution sections 7 of element processors calculate (19) and (20) using necessary data transmitted by the transmitters 50 of other clusters. These operations are called back-substitution.

The operation of the main controller G_{pc} 45 is described below with reference to Fig. 14, which shows a flow chart of its control algorithm at the level of above definition.

The first step distributes and assigns the rows of the coefficient matrix $A^{(0)}$ and the components of $b^{(0)}$ and x to the cluster CL_0 42,. . . , CL_u 43, . . . , CL_{P-1} 44 in such a manner as each block of k rows and corresponding 2k components (n - [n/k]k rows and 2(n - [n/k]k) components in the final distribution) are transmitted to the memory 1 of one node at a time in the cyclic order of $CL_0, \ldots, CL_{P-1}, CL_0, CL_1, \ldots$.

The next step tests if n is a multiple of k. If it is, then the next step initializes p as p = 0 and enters the loop of the left side. The t-th step within this loop orders the execution of the parallel preprocessing CLA_t for the (pk + t)th row of the current reduced matrix $A^{(pk+t-1)}$.The next step tests if p = n/k - 1. If it is, then the next step escapes the loop. If p < n/k - 1, then the next step orders the execution of the parallel updating B_c. The next step increments p by 1 and returns to the execution of the parallel preprocessing CLA_1.

If n is not a multiple of k, then the next step initializes p as p = 0 and enters the loop of the right side. Within this loop, the operations are the same except the fact that the condition for escaping the loop is p = [n/k], and the position of the testing for escape is between the parallel preprocessing $CLA_{n-[n/k]k}$ and $CLA_{n-[n/k]k+1}$.

After escaping one of the loops the final step orders the execution of back-substitution and terminates the whole operation to obtain the unknown vector x.

Fig. 15 is a block diagram of a parallel linear calculating equipment according to the sixth embodiment of the present invention In Fig. 15, 61 is a network; 62, 63, 64 are clusters CL_0, CL_u, CL_{P-1} mutually connected by the network 61; 65 is a main controller J_{pc} connected to each cluster. Fig. 16 is a block diagram of a cluster in Fig. 15. In Fig. 16, 1 is a memory; 46 is a C gateway that is a junction with the outside; 66, 67, 68 are element processors PE_1, PE_2,

$$PE_{P_c},$$

each connected to the memory 1; 50 is a transmitter that transmits data between the memory 1 and the outside through the C gateway 46. Fig. 17 is a block diagram of an element processor shown in Fig. 16. In Fig. 17, 2 is a pivoting section; 3, 4, 5 are preprocessing sections A_1, A_t, A_k respectively, each connected to the pivoting section 2; 9 is an updating section B' connected to the pivoting section 2; 10, 11, 12 are postprocessing sections C_1, C_t, C_{k-1} respectively, each connected to the pivoting section 2; 51 is a gateway that is a junction with the outside; 103 is a register set composed of k registers; 104 is an arithmetic unit.

Following is a description of the operation of each component of the fourth embodiment.

If the (pk + 1)th through (p + 1)k-th rows of $A^{(0)}$ and corresponding components of $b^{(0)}$ and x are assigned to the cluster CL_u 63, then the pivoting section 2 and the updating section B' 9 of each element processor of CL_u 63 take charge of part of the k rows and 2k components row by row, while the preprocessing section A_t 4 and postprocessing section C_t 11 of each element processor of CL_u 63 take charge of elements of the (pk + t)th row of $A^{(r)}$ and the (pk + t)th component of $b^{(r)}$ one by one.

Specifically, the pivoting section 2 of the element processor PE_1 of CL_u 63 determines the transposed pivot (3) of the (pk + 1)th row, and the preprocessing sections A_1 3 of element processors of CL_u 63 simultaneously calculate (4) and (5) for $pk + 2 \leq j \leq n$ with each A_1 3 calculating for elements and components in its charge, and the transmitter 50 transmits the results to the memory 1 of every other cluster through the C gateway 46, while the updating section B' 9 of the element processor in charge of the i-th row calculates (14) for every i such that $(p + 1)k + 1 \leq i \leq n$. This series of operations is below called parallel preprocessing CLA_1.

The preprocessing sections A_t 4 of the above cluster CL_u 63 simultaneously calculate (6), (7), (8), (9), (10) for $pk + t \leq j \leq n$ with each A_t 4 calculating for elements and components in its charge and,

immediately after the pivoting section 2 of the element processor PE_t of CL_u 63 determines the pivot (11), simultaneously calculate (12) and (13) for $pk + t + 1 \leq j \leq n$, and the transmitter 50 transmits the results to the memory 1 of every other cluster through the C gateway 46, while the updating section B' 9 of the element processor in charge of the i-th row calculates (30) for every i such that $(p + 1)k + 1 \leq i \leq n$. This series of operations is below called parallel preprocessing CLA_t, where $2 \leq t \leq k$.

The updating section B' 9 of each element processor in charge of the i-th row such that $1 \leq i \leq pk$ or $(p + 1)k + 1 \leq i \leq n$ if n is a multiple of k or $p < [n/k]$ and $1 \leq i \leq [n/k]k$ otherwise also calculates (14) through (18) for $(p + 1)k + 1 \leq j \leq n$ if n is a multiple of k or $p < [n/k]$ and for $[n/k]k + 1 \leq j \leq n$ otherwise, retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in the register set. These operations are below called parallel updating B'_c.

The postprocessing sections C_t 11 of element processors of the above CL_u 63 simultaneously calculate (21), (22), . . . , (29) for j such that $pk + t + 2 \leq j \leq n$ for $t = 1, 2, \ldots, k - 1$ if n is a multiple of k or $p < [n/k]$ and for $t = 1, 2, \ldots, n - [n/k]k$ otherwise with each C_t 11 calculating for elements and components in its charge. This series of operations is below called post-elimination C_c.

The operation of the main controller J_{pc} 65 is described below with reference to Fig. 18, which shows a flow chart of its control algorithm at the level of above definition.

The first step distributes and assigns the rows of the coefficient matrix $A^{(0)}$ and the components of $b^{(0)}$ and x to the clusters CL_0 62, . . . , CL_u 63, . . . , CL_{P-1} 64 in such a manner as each block of k rows and corresponding 2k components (n - [n/k]k rows and 2(n - [n/k]k) components in the final distribution) are transmitted to the memory 1 of one node at a time in the cyclic order of $CL_0, \ldots, CL_{P-1}, CL_0, CL_1, \ldots$.

The next step tests if n is a multiple of k. If it is, then the next step initializes p as $p = 0$ and enters the loop of the left side. The t-th step within this loop orders the execution of the parallel preprocessing CLA_t for the (pk + t)th row of the current reduced matrix $A^{(pk+t-1)}$.The next step orders the execution of the parallel updating B'_c. The next step orders the execution of the post-elimination C_c. The next step tests if $p = n/k - 1$. If it is, then the next step escapes the loop. If $p < n/k - 1$, then the next step increments p by 1 and returns to the execution of the parallel preprocessing CLA_1.

If n is not a multiple of k, then the next step initializes p as $p = 0$ and enters the loop of the right side. Within this loop, the operations are the same except the fact that the condition for escaping the loop is $p = [n/k]$, and if $p = [n/k]$, the steps skip the order for the execution of the parallel preprocessing $CLA_{n-[n/k]k+1}$ through CLA_k.

By the above processing, the unknown vector is obtained.

Fig. 19 shows a block diagram of an element processor or processor module of a parallel computer that implements the seventh embodiment of the present invention. In Fig 19, 201 is a gate way; 202 is a cache memory; 203 is a central processing unit; 204 is a local memory; 205 is a shared buss. Fig 20 shows a block diagram of a cluster composed of element processors 212, 213, . . . , 214, a C gateway 210, and a shared memory 211. A network of the parallel computer connects each of the clusters to each other, so that data can be transmitted between any two clusters. Let the number of element processors in each cluster be P_c and the total number of clusters be C. Then the total number P of element processors in the parallel computer is C^*P_c. Furthermore, let the clusters be denoted by $CL_1, CL_2, \ldots, CL_C$, and let the element processors of CL_u be denoted by $PR_{u\,1}, \ldots,$

$$PR_{u\ P_c}.$$

Fig. 21 is a block diagram of parallel linear computation method according to the seventh embodiment of the present invention implemented by a parallel computer structured above. In Fig. 21, 220 is a data distribution means; 221 is a pivoting means; 222 is an elementary pre-elimination means; 223 is a multi-pivot elimination means; 224 is an elimination testing means; 225 is a remainder elimination means; 226 is an elementary back-substitution mean; 227 is an elementary back-transmission means; 228 is an elementary back-calculation means; 229 is a back-processing testing means.

The operation of the parallel linear computation method of the seventh embodiment is described below with reference to Fig. 21.

In the first step, the data distribution means 220 distributes each i-th row of $A^{(0)}$ and i-th component of $b^{(0)}$ and x to the cluster CL_u such that $u = [i/P_c] - [[i/P_c]/C]C + 1$. Then the data distribution means 220 assigns each i-th row of $A^{(0)}$ and i-th component of $b^{(0)}$ distributed to the cluster CL_u to the element processor $PR_{u\,v}$ such that $v = i - [i/P_c]P_c + 1$. Then the data distribution means initializes k as $k = 0$.

In the second step, the elimination testing means 224 tests if the multi-pivot elimination means repeated its operation $[n/P_c]$ times, that is, whether $k = [n/P_c]$. If it did, then the process jumps to the fifth step. If it did not, the process goes to the third step.

In the third step, the elementary pre-elimination means 222 executes preliminary processing for the i-th rows of reduced matrices and the corresponding known vectors such that $i = kP_c + l$ and $l = 1, \ldots, P_c$ in this order. The processing involves a pivot choosing process for each l.

Methods of choosing a pivot are in general classified into either partial pivoting or full pivoting. Partial pivoting chooses as a pivot in each reduced matrix $A^{(r)}$ an element with the largest absolute value in the relevant column or row. Full pivoting chooses as a pivot in each reduced matrix $A^{(r)}$ an element with the largest absolute value in the submatrix of the columns or rows which have not hitherto been pivotal. Besides, if precision is not important so much, then choosing of a pivot is necessary only when the relevant diagonal element is 0, and in that case any nonzero element can be chosen as a pivot in partial pivoting. Pivoting methods in the present invention employ partial pivoting, and the present first method chooses the first nonzero element in the relevant row, and the present second method chooses an element with the greatest absolute value in the relevant row.

Fig. 23 shows the process of the pivot choosing means 221. In Fig. 23, 240 is a search means; 241 is a column number broadcasting means; 242 is an element interchange means; 243 is an component interchange means.

In the present first method of pivot choosing, the element processor in charge of each i-th row, by the search means 240, tests if $a^{(l-1)}_{l\,l} = 0$. If it is not, then the process terminates. If it is, then the element processor, by the search means 240, searches for a nonzero element in the i-th row of $A^{(l-1)}$ from $a^{(l-1)}_{l\,l+1}$ to $a^{(l-1)}_{i\,n}$ in this order. If $a^{(l-1)}_{i\,h}$ is the first such element, then the element processor, by the broadcasting means 241, notifies each element processor of the column number h by a broadcast. Specifically, the element processor either transmits h to a specified word of the shared memory 211 of each cluster, and each element processor refers to the word, or the element processor transmits h to a dedicated bus line, and each element processor fetches h into its local memory 204. Then each element processor, by the element interchange means 242, simultaneously interchanges the element with the column number i with the element with the column number h in the row in its charge. Then two element processors in charge of the i-th component and the h-th component of the unknown vector x respectively interchange these component by the component interchange means 243. The pivot choosing process terminates hereby.

In the present second method of pivot choosing, the element processor in charge of each i-th row, by the search means 240, sets $Max = |a^{(l-1)}_{l\,l}|$ and $Col = i$. The element processor then compares max with $|a^{(l-1)}_{l\,j}|$ for $j = i + 1, \ldots, n$ in this order and updates Max and Col as $Max = |a^{(l-1)}_{i\,j}|$ and $Col = j$, only if $|a^{(l-1)}_{l\,j}|$ is greater than Max. Then the element processor notifies each element processor of Col by a broadcast. The remaining steps are the same as above.

In the process of the elementary pre-elimination means 222, if $l = 1$, then the element processor $PR_{u\,1}$ incharge of the $(kP_c + 1)$th row in the cluster CL_u, where $u = k - [k/C] + 1$, calculates (32) and (33), and transmits the results to the shared memory of every other cluster to which the element processor in charge of an i-th row such that $kP_c + 2 \leq i \leq n$ belongs. If $2 \leq l \leq P_c$, then each element processor in charge of the i-th row such that $kP_c + l \leq i \leq n$ calculates (34), and the element processor $PR_{u\,l}$ calculates (35) and (36). Then after the pivot choosing means determines the pivot (37), the element processor $PR_{u\,l}$ calculates (38) and (39) and transmits the results to the shared memory of every other cluster to which the element processor in charge of an i-th row such that $kP_c + l + 1 \leq i \leq n$ belongs.

In the fourth step, by the multi-pivot elimination means 223, each element processor in charge of the i-th row such that $(k + 1)P_c + 1 \leq i \leq n$ calculate (40) and (41) for i.

In the fifth step, by the remainder elimination means 225, each element processor in charge of the i-th row such that $[n/P_c]P_c + 1 \leq i \leq n$ executes the same operation as in the elementary pre-elimination means 232 for $l = 2, \ldots, n - [n/P_c]P_c$. Then this step initializes i as $i = n$, and goes to the sixth step.

In the sixth step, by the elementary back-substitution means 226, the element processor in charge of the i-th row calculates (42).

In the seventh step, the back-processing testing means 229 tests if $i = n$. If it is, then the solution of the system of linear equation (2) has been obtained by the above elementary back-substitution as (44), and the process terminates. If it is not, then the process proceeds to the eighth step.

In the eighth step, an elementary back-transmission means that transmits x_i to the shared memory of

every clusters such that the element processor in charge of an h-th row such that $1 \leq h \leq i - 1$ belongs.

In the ninth step, by the elementary back-calculation means, each element processor in charge of the h-th row such that $1 \leq h \leq i - 1$ calculates (43). Then this step decrements i by 1, and increments goes to the sixth step.

Fig. 22 is a block diagram of parallel linear calculating method in the eighth embodiment of the present invention implemented by a parallel computer structured as in the seventh embodiment. In Fig. 22, 220 is a data distribution means; 221 is a pivot choosing means; 231 is an elimination testing means, 232 is an elementary pre-elimination means; 233 is a multi-pivot elimination means; 234 is an elementary post-elimination means; 225 is a post-elimination processing means; 236 is a remainder elimination means.

The operation of the parallel linear computation method of the seventh embodiment is described below with reference to Fig. 22.

In the first step, the data distribution means 220 distributes each i-th row of $A^{(0)}$ and i-th component of $b^{(0)}$ and x to the cluster CL_u such that $u = [i/P_c] - [[i/P_c]/C]C + 1$. Then the data distribution means 220 assigns each i-th row of $A^{(0)}$ and i-th component of $b^{(0)}$ distributed to the cluster CL_u to the element processor $PR_{u\,v}$ such that $v = i - [i/P_c]P_c + 1$. Then the data distribution means initializes k as $k = 0$.

In the second step, the elimination testing means 231 tests if the multi-pivot elimination means repeated its operation $[n/P_c]$ times, that is, whether $k = [n/P_c]$. If it did, then the process jumps to the sixth step. If it did not, the process goes to the third step.

In the third step, the elementary pre-elimination means 232 executes preliminary processing for the i-th rows of the reduced matrices and the corresponding known vectors such that $i = kP_c + l$ and $l = 1, \ldots, P_c$ in this order. The processing involves a pivot choosing process for each l, which is the same as in the seventh embodiment.

In the pre-elimination means 232, if $l = 1$, then after the pivot choosing means 221 determines the pivot (31), the element processor $PR_{u\,1}$ in charge of the $(kP_c + 1)$th row in the cluster CL_u, where $u = k - [k/C] + 1$, calculates (32) and (33), and transmits the results to the shared memory of every other cluster. If $2 \leq l \leq P_c$, then each element processor in charge of the i-th row such that $kP_c + l \leq i \leq n$ calculates (34), and the element processor $PR_{u\,l}$ calculates (35) and (36). Then after the pivot choosing means determines the pivot (37), the element processor $PR_{u\,l}$ calculates (38) and (39) and transmits the results to the shared memory of every other cluster.

In the fourth step, by the multi-pivot elimination means 233, each element processor in charge of the i-th row such that $1 \leq i \leq kP_c$ or $(k + 1)P_c + 1 \leq i \leq n$ calculates (40) and (41).

In the fifth step, the post-elimination processing means 235 eliminates unnecessary elements generated by the multi-pivot elimination means 233. The core of the post-elimination processing means 235 is the elementary post-elimination means 234, which calculates (45) and (46) in the element processor in charge of the i-th row.

By the post-elimination processing means the element processor in charge of the $(kP_c + w)$th row calculates (45) and (46), where $i = P_c + w$ and $l = -w + q + 1$, from $w = 1$ to $w = q$ for $q = 1, 2, \ldots, P_c - 1$.

In the sixth step, by the remainder elimination means 236, each element processor in charge of the i-th row such that $[n/P_c]P_c + 1 \leq i \leq n$ executes the operation of the elementary pre-elimination means 232. Then the remainder elimination means executes operation of the multi-pivot elimination means 233 followed by the post-elimination processing means 235. The operation of pre-elimination processing means 232 should be executed for $l = 1, \ldots, n - [n/P_c]P_c$. The operation of the multi-pivot elimination means 233 should be executed by calculating (40) and (41) for $1 \leq i \leq [n/P_c]P_c$ and $k = [n/P_c]$. The operation of the post-elimination processing means 235 should be executed from $q = 1$ to $q = n - [n/P_c]P_c$ for $k = [n/P_c]$.

The unknown vector x is obtained as the vector $b^{(r)}$ after the above operation.

If the preprocessing section A_t and the postprocessing section C_t have their own register sets as the updating section B and B' in the first embodiment through the six embodiment, and their operations are executed by retaining values of variables and divisors, then the number of load-and-store operations for the memory are reduced, and further improvement in computation speed can be achieved.

In the seventh and eighth embodiments two components of the unknown vector should be interchanged if the corresponding columns are interchanged by the pivoting means. However, it is not necessary to actually transpose the components. By simply memorizing the correct position of the components after each interchange of columns, the correct solution is obtained by considering the positions in the final substitution to the components of the unknown vector.

Thus the present invention provides high-speed linear calculating equipment and parallel linear calculating equipment for solving systems of linear equations by means of Gauss's elimination method and Gauss-Jordan's method based on multi-pivot simultaneous elimination and scalar operations. The speed-up

is achieved by reducing the number of load-and-store operations for the memory by retaining values of variables in register sets in updating processing, and reducing the number of iteration by multi-pivot simultaneous elimination. And the present invention is easily implementation in scalar computers. In fact, an experiment done in a scalar computer by means of software showed that Gauss's method and Gauss-Jordan's method based on 8-pivot simultaneous elimination was 2.5 times faster than original Gauss's elimination method and Gauss-Jordan's elimination method.

As for the parallel calculating equipment of the third through sixth embodiments of the seventh and eighth embodiments, each memory is assigned blocks of k rows of the coefficient matrix $A^{(0)}$ for the k-pivot simultaneous elimination method, so that effects of parallel computation are enhanced. In the fifth and sixth embodiments, where element processors are clustered, the preprocessing or both the preprocessing and the postprocessing are also made parallel, and the computation is more effective. In these embodiments, a theoretical estimation has shown that if the number of components of the unknown vector X is sufficiently large for a definite number of processors, then the effects of parallel computation are sufficiently powerful. Therefore, parallel linear calculating equipment effectively employing Gauss method and Gauss-Jordan method based on multi-pivot simultaneous elimination has been obtained.

Furthermore, the present invention effectively makes possible high-speed parallel computation for solving systems of linear equations using a parallel computer with a number of element processors by means of the methods of the seventh and eighth embodiments.

Although the present invention has been fully described in connection with the preferred embodiments thereof with reference to the accompanying drawings, it is to be noted that various changes and modifications are apparent to those skilled in the art. Such changes and modifications are to be understood as included within the scope of the present invention as defined by the appended claims unless they depart therefrom.

Claims

1. Linear calculating equipment comprising:

a memory that stores a plurality of coefficient matrices $A^{(r)}$ and vectors $b^{(r)}$ and an unknown vector x expressed by

$$A^{(r)} = (a_{ij}^{(r)}) \ , \ 1 \leq i, j \leq n,$$

$$b^{(r)} = (b_1^{(r)}, b_2^{(r)}, \ldots, b_n^{(r)})^t,$$

$$x = (x_1, x_2, \ldots, x_n)^t$$

where r = 0, 1, . . . , n, for a given system of linear equations

$$A^{(0)}x = b^{(0)},$$

a pivoting section connected to said memory for choosing a pivot in the i-th row of $A^{(i-1)}$, and interchanges the i-th column with the chosen pivotal column for $1 \leq i \leq n$,

a preprocessing section A_1 which, immediately after said pivot choosing section determines the transposed pivot

$$a_{pk+1\,pk+1}^{(pk)}$$

for a given positive integer k and a value of a nonnegative integral parameter p, calculates

$$a_{pk+1j}^{(pk+1)} = a_{pk+1j}^{(pk)} / a_{pk+1pk+1}^{(pk)}$$

for pk + 2 ≤ j ≤ n and

$$b_{pk+1}^{(pk+1)} = b_{pk+1}^{(pk)} / a_{pk+1pk+1}^{(pk)},$$

a plurality of preprocessing sections A_t, where t = 2, 3, . . . , k, each of which is connected to said memory and calculates

$$Reg_{pk+t}^{(0)} = a_{pk+tpk+1}^{(pk)},$$

$$Reg_{pk+t}^{(1)} = a_{pk+tpk+2}^{(pk)} - Reg_{pk+t}^{(0)} a_{pk+1pk+2}^{(pk+1)},$$

. . . ,

$$Reg_{pk+t}^{(t-2)} = a_{pk+tpk+t-1}^{(pk)} - \sum_{m=1}^{t-2} Reg_{pk+t}^{(m-1)} a_{pk+mpk+t-1}^{(pk+m)},$$

$$a_{pk+tj}^{(pk+t-1)} = a_{pk+tj}^{(pk)} - \sum_{m=1}^{t-1} Reg_{pk+t}^{(m-1)} a_{pk+mj}^{(pk+m)},$$

$$b_{pk+t}^{(pk+t-1)} = b_{pk+t}^{(pk)} - \sum_{m=1}^{t-1} Reg_{pk+t}^{(m-1)} b_{pk+m}^{(pk+m)}$$

for pk + t ≤ j ≤ n, and, immediately after said pivot choosing section determines the transposed pivot

$$a_{pk+tpk+t}^{(pk+t-1)},$$

calculates

$$a_{pk+tj}^{(pk+t)} = a_{pk+tj}^{(pk+t-1)} / a_{pk+tpk+t}^{(pk+t-1)},$$

$$b_{pk+t}^{(pk+t)} = b_{pk+t}^{(pk+t-1)} / a_{pk+tpk+t}^{(pk+t-1)}$$

for pk + t + 1 ≤ j ≤ n,

an updating section which is connected to said memory, comprises a register set of k registers and an arithmetic unit, and calculates

$$Reg_i^{(0)} = a_{ipk+1}^{(pk)},$$

$$Reg_i^{(1)} = a_{ipk+2}^{(pk)} - Reg_i^{(0)} a_{pk+1pk+2}^{(pk+1)},$$

. . . . .

$$Reg_i^{(k-1)} = a_{i(p+1)k}^{(pk)} - \sum_{m=1}^{k-1} Reg_i^{(m-1)} a_{pk+m(p+1)k}^{(pk+m)},$$

$$a_{ij}^{((p+1)k)} = a_{ij}^{(pk)} - \sum_{m=1}^{k} Reg_i^{(m-1)} a_{pk+mj}^{(pk+m)},$$

$$b_i^{((p+1)k)} = b_i^{(pk)} - \sum_{m=1}^{k} Reg_i^{(m-1)} b_{pk+m}^{(pk+m)}$$

for $(P+1)k + 1 \leq i, j \leq n$ retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in said register set,

a back-substitution section which is connected to said memory and obtains the value of said unknown vector x by calculating

$$x_i = b_i^{(n)}$$

and

$$b_h^{(n+h-i+1)} = b_h^{(n+h-i)} - a_{hi}^{(h)} x_i$$

for $1 \leq h \leq i - 1$ for $i = n, n - 1, \ldots, 1$ in this order of i, and

a main controller G which, if n is a multiple of said k, instructs said pivot choosing section, said preprocessing sections $A_1, \ldots, A_k$, and said updating section to repeat their operations for $p = 0, 1, \ldots, n/k - 2$, and instructs said pivot choosing section and said preprocessing sections $A_1, \ldots, A_k$ to execute their operations for $p = n/k - 1$, and, if n is not a multiple of said k, instructs said pivot choosing

section, said preprocessing sections $A_1, \ldots, A_k$, and said updating section B to repeat their operations for p = 0, 1, . . . [n/k] - 1, where [s] denotes the greatest integer not exceeding s, and instructs said pivot choosing section and said preprocessing sections $A_1, \ldots, A_{n-[n/k]k}$ to execute their operations, and in both cases, instructs said back substitution section to obtain said unknown vector.

2. Linear calculating equipment comprising:

a memory which stores a plurality of coefficient matrices $A^{(r)}$ and vectors $b^{(r)}$ and an unknown vector x expressed by

$$A^{(r)} = (a_{ij}^{(r)}) , \ 1 \leq i, j \leq n,$$

$$b^{(r)} = (b_1^{(r)}, b_2^{(r)}, \ldots, b_n^{(r)})^t,$$

$$x = (x_1, x_2, \ldots, x_n)^t$$

where r = 0, 1, . . . , n, for a given system of linear equations

$$A^{(0)}x = b^{(0)},$$

a pivot choosing section which is connected to said memory, chooses a pivot in the i-th row of $A^{(i-1)}$, and interchanges the i-th column with the chosen pivotal column for $1 \leq i \leq n$,

a preprocessing section A_1 which, immediately after said pivot choosing section determines the transposed pivot

$$a_{pk+1\,pk+1}^{(pk)}$$

for a given positive integer k and a value of a nonnegative integral parameter p, calculates

$$a_{pk+1\,j}^{(pk+1)} = a_{pk+1\,j}^{(pk)} / a_{pk+1\,pk+1}^{(pk)}$$

for $pk + 2 \leq j \leq n$ and

$$b_{pk+1}^{(pk+1)} = b_{pk+1}^{(pk)} / a_{pk+1\,pk+1}^{(pk)},$$

k - 1 preprocessing sections A_t, where t = 2, 3, . . . , k, each of which is connected to said memory and calculates

$$Reg_{pk+t}^{(0)} = a_{pk+tpk+1}^{(pk)},$$

$$Reg_{pk+t}^{(1)} = a_{pk+tpk+2}^{(pk)} - Reg_{pk+t}^{(0)} a_{pk+1pk+2}^{(pk+1)},$$

. . . ,

$$Reg_{pk+t}^{(t-2)} = a_{pk+tpk+t-1}^{(pk)} - \sum_{m=1}^{t-2} Reg_{pk+t}^{(m-1)} a_{pk+mpk+t-1}^{(pk+m)},$$

$$a_{pk+tj}^{(pk+t-1)} = a_{pk+tj}^{(pk)} - \sum_{m=1}^{t-1} Reg_{pk+t}^{(m-1)} a_{pk+mj}^{(pk+m)},$$

$$b_{pk+t}^{(pk+t-1)} = b_{pk+t}^{(pk)} - \sum_{m=1}^{t-1} Reg_{pk+t}^{(m-1)} b_{pk+m}^{(pk+m)}$$

for $pk + t \leq j \leq n$, and, immediately after said pivot choosing section determines the transposed pivot

$$a_{pk+tpk+t}^{(pk+t-1)},$$

calculates

$$a_{pk+tj}^{(pk+t)} = a_{pk+tj}^{(pk+t-1)} / a_{pk+tpk+t}^{(pk+t-1)},$$

$$b_{pk+t}^{(pk+t)} = b_{pk+t}^{(pk+t-1)} / a_{pk+tpk+t}^{(pk+t-1)}$$

for $pk + t + 1 \leq j \leq n$,

an updating section which is connected to said memory, comprises a register set of k registers and an arithmetic unit, and calculates

$$Reg_i^{(0)} = a_{ipk+1}^{(pk)},$$

$$Reg_i^{(1)} = a_{ipk+2}^{(pk)} - Reg_i^{(0)} a_{pk+1pk+2}^{(pk+1)},$$

. . . . .

$$Reg_i^{(k-1)} = a_{i(p+1)k}^{(pk)} - \sum_{m=1}^{k-1} Reg_i^{(m-1)} a_{pk+m(p+1)k}^{(pk+m)},$$

$$a_{ij}^{(p+1)k} = a_{ij}^{(pk)} - \sum_{m=1}^{k} Reg_i^{(m-1)} a_{pk+mj}^{(pk+m)},$$

$$b_i^{(p+1)k} = b_i^{(pk)} - \sum_{m=1}^{K} Reg_i^{(m-1)} b_{pk+m}^{(pk+m)}$$

for $1 \le i \le pk$, $(p + 1)k + 1 \le i \le n$, and $(p + 1)k + 1 \le j \le n$ if n is a multiple of k or $p < [n/k]$ and for $1 \le i \le [n/k]k$, $[n/k]k + 1 \le j \le n$ otherwise, where [s] denotes the greatest integer not exceeding s, retaining the values
of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in said register set,

k - 1 postprocessing sections C_t, where $t = 1, 2, \ldots, k - 1$, each of which is connected to said memory and calculates

$$Reg^{(0)} = a_{pk+1pk+t+1}^{(pk+t)},$$

$$Reg^{(1)} = a_{pk+2pk+t+1}^{(pk+t)},$$

. . . ,

$$Reg^{(t-1)} = a_{pk+tpk+t+1}^{(pk+t)},$$

$$a_{pk+1j}^{(pk+t+1)} = a_{pk+1j}^{(pk+t)} - Reg^{(0)}\, a_{pk+t+1j}^{(pk+t+1)},$$

$$a_{pk+2j}^{(pk+t+1)} = a_{pk+2j}^{(pk+t)} - Reg^{(1)}\, a_{pk+t+1j}^{(pk+t+1)},$$

$$\cdots,$$

$$a_{pk+tj}^{(pk+t+1)} = a_{pk+tj}^{(pk+t)} - Reg^{(t-1)}\, a_{pk+t+1j}^{(pk+t+1)},$$

$$b_{pk+1}^{(pk+t+1)} = b_{pk+1}^{(pk+t)} - Reg^{(0)}\, b_{pk+t+1}^{(pk+t+1)},$$

$$b_{pk+2}^{(pk+t+1)} = b_{pk+2}^{(pk+t)} - Reg^{(1)}\, b_{pk+t+1}^{(pk+t+1)},$$

$$\cdots\cdots$$

$$b_{pk+t}^{(pk+t+1)} = b_{pk+t}^{(pk+t)} - Reg^{(t-1)}\, b_{pk+t+1}^{(pk+t+1)}$$

for $pk + t + 2 \leq j \leq n$,

a main controller that, if n is a multiple of k, instructs said pivot choosing section, said preprocessing sections $A_1, \ldots, A_k$, said updating section B', and said postprocessing sections $C_1, \ldots, C_{k-1}$ to repeat their operations for $p = 0, 1, \ldots, n/k - 1$, and, if n is not a multiple of k, instructs said pivot choosing section, said preprocessing sections $A_1, \ldots, A_k$, the updating section, and said postprocessing sections $C_1, \ldots, C_{k-1}$ to repeat their operations for $p = 0, 1, \ldots [n/k] - 1$, and instructs said pivot choosing section, said preprocessing sections $A_1, \ldots, A_{n-[n/k]k}$, said updating section B', and said postprocessing sections $C_1, \ldots, C_{n-[n/k]k}$ to execute their operations for $p = [k/n]$.

3. Linear calculating equipment comprising:

a network,

a main controller which is connected to said network and executes flow control,

a plurality of nodes α_u, where $u = 0, 1, \ldots, P - 1$, each of which is connected to each other by said network and comprises:

a memory which stores blocks of k rows of each coefficient matrix $A^{(r)}$ and corresponding k components of each known vector $b^{(r)}$ and an unknown vector x assigned to said node and expressed by

$$A^{(r)} = (a_{ij}^{(r)}),\ 1 \leq i, j \leq n,$$

$$b^{(r)} = (b_1^{(r)}, b_2^{(r)}, \ldots, b_n^{(r)})^t,$$

$$x = (x_1, x_2, \ldots, x_n)^t$$

for $0 \leq r \leq n$, for a given system of linear equations

$A^{(0)}x = b^{(0)}$,

a pivot choosing section which is connected to said memory, chooses a pivot in the i-th row of $A^{(i-1)}$, and interchanges the i-th column with the chosen pivotal column for i in said node's charge,

a preprocessing section A_1 which, immediately after said pivot choosing section determines the transposed pivot

$$a_{pk+1\,pk+1}^{(pk)} \tag{47}$$

for a given positive integer k and a value of a nonnegative integral parameter p, calculates

$$a_{pk+1\,j}^{(pk+1)} = a_{pk+1\,j}^{(pk)} / a_{pk+1\,pk+1}^{(pk)} \tag{48}$$

for $pk + 2 \leq j \leq n$ and

$$b_{pk+1}^{(pk+1)} = b_{pk+1}^{(pk)} / a_{pk+1\,pk+1}^{(pk)} \tag{49}$$

for pk + 1 in said node's charge,

k - 1 preprocessing sections A_t, where t = 2, 3, . . . , k, each of which is connected to said memory and calculates

$$Reg_{pk+t}^{(0)} = a_{pk+t\,pk+1}^{(pk)}, \tag{50}$$

$$Reg_{pk+t}^{(1)} = a_{pk+t\,pk+2}^{(pk)} - Reg_{pk+t}^{(0)} a_{pk+1\,pk+2}^{(pk+1)}, \tag{51}$$

. . . ,

$$Reg_{pk+t}^{(t-2)} = a_{pk+t\,pk+t-1}^{(pk)} - \sum_{m=1}^{t-2} Reg_{pk+t}^{(m-1)} a_{pk+m\,pk+t-1}^{(pk+m)}, \tag{52}$$

$$a_{pk+t\,j}^{(pk+t-1)} = a_{pk+t\,j}^{(pk)} - \sum_{m=1}^{t-1} Reg_{pk+t}^{(m-1)} a_{pk+m\,j}^{(pk+m)}, \tag{53}$$

$$b_{pk+t}^{(pk+t-1)} = b_{pk+t}^{(pk)} - \sum_{m=1}^{t-1} Reg_{pk+t}^{(m-1)} b_{pk+m}^{(pk+m)} \tag{54}$$

for $pk + t \leq j \leq n$ for pk + t in said node's charge, and, immediately after said pivot choosing section determines the transposed pivot

$$a_{pk+t\,pk+t}^{(pk+t-1)}, \tag{55}$$

calculates

$$a_{pk+t\,j}^{(pk+t)} = a_{pk+t\,j}^{(pk+t-1)} / a_{pk+t\,pk+t}^{(pk+t-1)}, \tag{56}$$

$$b_{pk+t}^{(pk+t)} = b_{pk+t}^{(pk+t-1)} / a_{pk+t\,pk+t}^{(pk+t-1)} \tag{57}$$

for $pk + t + 1 \leq j \leq n$ for $pk + t$ in said node's charge,

an updating section which is connected to said memory, comprises a register set of k registers and an arithmetic unit, and calculates

$$Reg_i^{(0)} = a_{i\,pk+1}^{(pk)}, \tag{58}$$

$$Reg_i^{(1)} = a_{i\,pk+2}^{(pk)} - Reg_i^{(0)}\, a_{pk+1\,pk+2}^{(pk+1)}, \tag{59}$$

. . . . .

$$Reg_i^{(k-1)} = a_{i\,(p+1)k}^{(pk)} - \sum_{m=1}^{k-1} Reg_i^{(m-1)}\, a_{pk+m\,(p+1)k}^{(pk+m)}, \tag{60}$$

$$a_{ij}^{((p+1)k)} = a_{ij}^{(pk)} - \sum_{m=1}^{k} Reg_i^{(m-1)}\, a_{pk+m\,j}^{(pk+m)}, \tag{61}$$

$$b_i^{((p+1)k)} = b_i^{(pk)} - \sum_{m=1}^{k} Reg_i^{(m-1)}\, b_{pk+m}^{(pk+m)} \tag{62}$$

for $(p + 1)k + 1 \leq j \leq n$ for i in said node's charge retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in said register set,

a back-substitution section which is connected to said memory and obtains the value of said unknown vector x by calculating

$$x_i = b_i^{(n)} \tag{63}$$

and

$$b_h^{(n+h-i+1)} = b_h^{(n+h-i)} - a_{hi}^{(h)} x_i \qquad (64)$$

for i and h in said node's charge,

a gateway which is connected to said memory and is a junction with the outside, and

a transmitter which is connected to said memory and transmits data between said memory and the outside of said node through said gateway.

4. The equipment as claimed in claim 3, further including:

parallel preprocessing A_1 wherein if the (pk + 1)th through (p + 1)k-th rows of said $A^{(0)}$ and corresponding components of $b^{(0)}$ and x are assigned to said node α_u, then said pivot choosing section of said node α_u determines the pivot (47), and said preprocessing section of said node α_u calculates (48) and (49) for $pk + 2 \leq j \leq n$, and said transmitter transmits the results to said memory of said every other node through said gateway, while said updating section B of said node in charge of the i-th row calculates (58) for every i such that $(p + 1)k + 1 \leq i \leq n$,

parallel preprocessing A_t, where t = 2, 3, . . . , k, wherein said preprocessing section A_t of said node α_u calculates (50) through (54) for $pk + t \leq j \leq n$, and, immediately after said pivot choosing section of α_u determines the pivot (55), calculates (56) and (57) for $pk + t + 1 \leq j \leq n$, and said transmitter transmits the results to said memory of said every other node through said gateway, while said updating section B of said node in charge of the i-th row calculates

$$Reg_i^{(t-1)} = a_{ipk+t}^{(pk)} - \sum_{m=1}^{k-1} Reg_i^{(m-1)} a_{pk+m\,pk+t}^{(pk+m)} \qquad (65)$$

for every i such that $(p + 1)k + 1 \leq i \leq n$,

parallel updating means wherein said updating section B of said each node in charge of the i-th row such that $(p + 1)k + 1 \leq i \leq n$ calculates (58) through (62) for $(p + 1)k + 1 \leq j \leq n$ retaining the values of

$$Reg_i^{(0)}, \ . \ . \ . \ , Reg_i^{(k)}$$

in said register set, and

back-substitution wherein said back-substitution section of said each node in charge of the i-th and h-th components calculates (63) and (64) for $1 \leq h \leq i - 1$ for i = n, n - 1, . . . , 1 in this order of i,

5. The equipment as claimed in claim 4 wherein said main controller distributes and assigns the rows of said coefficient matrix $A^{(0)}$ and the components of said $b^{(0)}$ and x to said nodes in such a manner as each block of consecutive said k rows and corresponding 2k components is transmitted to said memory of said one node in the cyclic order of said $\alpha_0, \ldots, \alpha_{P-1}, \alpha_0, \alpha_1 \ldots$, and, if n is a multiple of k, instructs said each node to execute said parallel preprocessing A_1 through A_k and said parallel updating B for p = 0, 1, . . . , n/k - 2 and to execute said parallel preprocessing A_1 through A_k for p = n/k - 1, and, if n is not a multiple of k, instructs said each node to execute said parallel preprocessing A_1 through A_k and said parallel updating B for p = 0, 1, . . . , [n/k] - 1, where [s] denotes the greatest integer not exceeding s, and to execute said parallel preprocessing A_1 through $A_{n-[n/k]k}$ for p = [n/k], and instructs said nodes to obtain said unknown vector x by executing back-substitution.

6. Linear calculating equipment comprising:

a network,

a main controller that is connected to said network and executes flow control, and

a plurality of nodes α_u, where u = 0, 1, . . . , P - 1, each of which is connected to each other by

said network and comprises:

a memory that stores a plurality of coefficient matrices $A^{(r)}$ and vectors $b^{(r)}$ and an unknown vector x expressed by

$$A^{(r)} = (a_{ij}^{(r)}) , \ 1 \leq i, j \leq n,$$

$$b^{(r)} = (b_1^{(r)}, b_2^{(r)}, \ldots, b_n^{(r)})^t,$$

$x = (x_1, x_2, \ldots, x_n)^t$

where r = 0, 1, . . . , n, for a given system of linear equations

$A^{(0)}x = b^{(0)}$,

a pivoting section that is connected to said memory, chooses a pivot in the i-th row of $A^{(i-1)}$, and interchanges the i-th column with the chosen pivotal column for i in said node's charge,

a preprocessing section A_1 that, immediately after said pivot choosing section determines the transposed pivot

$$a_{pk+1\,pk+1}^{(pk)} \qquad (66)$$

for a given positive integer k and a value of a nonnegative integral parameter p, calculates

$$a_{pk+1\,j}^{(pk+1)} = a_{pk+1\,j}^{(pk)} / a_{pk+1\,pk+1}^{(pk)} \qquad (67)$$

for pk + 2 ≤ j ≤ n and

$$b_{pk+1}^{(pk+1)} = b_{pk+1}^{(pk)} / a_{pk+1\,pk+1}^{(pk)} \qquad (68)$$

for pk + 1 in said node's charge,

k - 1 preprocessing sections A_t, where t = 2, 3, . . . , k, each of which is connected to said memory and calculates

$$Reg_{pk+t}^{(0)} = a_{pk+tpk+1}^{(pk)}, \tag{69}$$

$$Reg_{pk+t}^{(1)} = a_{pk+tpk+2}^{(pk)} - Reg_{pk+t}^{(0)} a_{pk+1pk+2}^{(pk+1)}, \tag{70}$$

. . . ,

$$Reg_{pk+t}^{(t-2)} = a_{pk+tpk+t-1}^{(pk)} - \sum_{m=1}^{t-2} Reg_{pk+t}^{(m-1)} a_{pk+mpk+t-1}^{(pk+m)}, \tag{71}$$

$$a_{pk+tj}^{(pk+t-1)} = a_{pk+tj}^{(pk)} - \sum_{m=1}^{t-1} Reg_{pk+t}^{(m-1)} a_{pk+mj}^{(pk+m)}, \tag{72}$$

$$b_{pk+t}^{(pk+t-1)} = b_{pk+t}^{(pk)} - \sum_{m=1}^{t-1} Reg_{pk+t}^{(m-1)} b_{pk+m}^{(pk+m)} \tag{73}$$

for $pk + t \le j \le n$ for $pk + t$ in said node's charge, and, immediately after said pivot choosing section determines the transposed pivot

$$a_{pk+tpk+t}^{(pk+t-1)}, \tag{74}$$

calculates

$$a_{pk+tj}^{(pk+t)} = a_{pk+tj}^{(pk+t-1)} / a_{pk+tpk+t}^{(pk+t-1)}, \tag{75}$$

$$b_{pk+t}^{(pk+t)} = b_{pk+t}^{(pk+t-1)} / a_{pk+tpk+t}^{(pk+t-1)} \tag{76}$$

for $pk + t + 1 \le j \le n$ for $pk + t$ in said node's charge,

an updating section which is connected to said memory, comprises a register set of k registers and an arithmetic unit, and calculates

$$Reg_i^{(0)} = a_{ipk+1}^{(pk)}, \tag{77}$$

$$Reg_i^{(1)} = a_{ipk+2}^{(pk)} - Reg_i^{(0)} a_{pk+1pk+2}^{(pk+1)}, \tag{78}$$

. . . . .

$$Reg_i^{(k-1)} = a_{i(p+1)k}^{(pk)} - \sum_{m=1}^{k-1} Reg_i^{(m-1)} a_{pk+m(p+1)k}^{(pk+m)}, \tag{79}$$

$$a_{ij}^{((p+1)k)} = a_{ij}^{(pk)} - \sum_{m=1}^{k} Reg_i^{(m-1)} a_{pk+mj}^{(pk+m)}, \tag{80}$$

$$b_i^{((p+1)k)} = b_i^{(pk)} - \sum_{m=1}^{k} Reg_i^{(m-1)} b_{pk+m}^{(pk+m)} \tag{81}$$

for $(p + 1)k + 1 \leq j \leq n$ for i in said node's charge retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in said register set, and

k - 1 postprocessing sections C_t, where $t = 1, 2, \ldots, k - 1$, each of which is connected to said memory and calculates

$$Reg^{(0)} = a_{pk+1pk+t+1}^{(pk+t)}, \tag{82}$$

$$Reg^{(1)} = a_{pk+2pk+t+1}^{(pk+t)}, \tag{83}$$

. . . ,

$$Reg^{(t-1)} = a_{pk+tpk+t+1}^{(pk+t)}, \tag{84}$$

$$a_{pk+1j}^{(pk+t+1)} = a_{pk+1j}^{(pk+t)} - Reg^{(0)} a_{pk+t+1j}^{(pk+t+1)}, \tag{85}$$

$$a_{pk+2j}^{(pk+t+1)} = a_{pk+2j}^{(pk+t)} - Reg^{(1)} a_{pk+t+1j}^{(pk+t+1)}, \tag{86}$$

. . . ,

$$a_{pk+tj}^{(pk+t+1)} = a_{pk+tj}^{(pk+t)} - Reg^{(t-1)}\, a_{pk+t+1j}^{(pk+t+1)}, \qquad (87)$$

$$b_{pk+1}^{(pk+t+1)} = b_{pk+1}^{(pk+t)} - Reg^{(0)}\, b_{pk+t+1}^{(pk+t+1)}, \qquad (88)$$

$$b_{pk+2}^{(pk+t+1)} = b_{pk+2}^{(pk+t)} - Reg^{(1)}\, b_{pk+t+1}^{(pk+t+1)}, \qquad (89)$$

. . . . .

$$b_{pk+t}^{(pk+t+1)} = b_{pk+t}^{(pk+t)} - Reg^{(t-1)}\, b_{pk+t+1}^{(pk+t+1)} \qquad (90)$$

for $pk + t + 2 \le j \le n$ for $pk + t$ in said node's charge,

a gateway which is connected to said memory and is a junction with the outside, and

a transmitter which is connected to said memory and transmits data between said memory and the outside of said node through said gateway.

7. The equipment as claimed in claim 6, further including:

parallel preprocessing A_1 wherein if the $(pk + 1)$th through $(p + 1)k$-th rows of said $A^{(0)}$ and corresponding components of said $b^{(0)}$ and x are assigned to said node α_u, then said pivot choosing section of said node α_u determines the pivot (66), and said preprocessing section of said node α_u calculates (67) and (68) for $pk + 2 \le j \le n$, and said transmitter transmits the results to said memory of said every other node through said gateway, while said updating section B' of said node in charge of the i-th row calculates (77) for every i such that $(p + 1)k + 1 \le i \le n$,

parallel preprocessing A_t, where $t = 2, 3, \ldots, t$, wherein said preprocessing section A_t of said node α_u calculates (69) through (73) for $pk + t \le j \le n$, and, immediately after said pivot choosing section of α_u determines the pivot (74), calculates (75) and (76) for $pk + t + 1 \le j \le n$, and said transmitter transmits the results to said memory of said every other node through said gateway, while said updating section B' of said node in charge of the i-th row calculates

$$Reg_i^{(t-1)} = a_{ipk+t}^{(pk)} - \sum_{m=1}^{k-1} Reg_i^{(m-1)}\, a_{pk+mpk+t}^{(pk+m)} \qquad (91)$$

for every i such that $(p + 1)k + 1 \le i \le n$,

parallel updating means wherein said updating section B' of said each node in charge of the i-th row such that $1 \le i \le pk$ or $(p + 1)k + 1 \le i \le n$ if n is a multiple of k or $p < [n/k]$, where [s] denotes the greatest integer not exceeding s, and $1 \le i \le [n/k]k$ otherwise calculates (77) through (81) for $(p + 1)k + 1 \le j \le n$ if n is a multiple of k or $p < [n/k]$ and for $[n/k]k + 1 \le j \le n$ otherwise, retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in said register set, and

post-elimination C wherein said postprocessing sections C_t of said node α_u calculates (82) through (90) for $pk + t + 2 \le j \le n$ for $t = 1, 2, \ldots, k - 1$ if n is a multiple of k or $p < [n/k]$ and for $t = 1, 2, \ldots, n - [n/k]k$.

8. The equipment as claimed in claim 7 wherein said main controller distributes the rows of said coefficient matrix $A^{(0)}$ and the components of said $b^{(0)}$ and x to said nodes in such a manner as each

block of consecutive k rows and corresponding 2k components is transmitted to said memory of said one node in the cyclic order of $\alpha_0, \ldots, \alpha_{P-1}, \alpha_0, \alpha_1, \ldots$, and, if n is a multiple of k, instructs said each node to execute said parallel preprocessing A_1 through A_k, said parallel updating B', and said post-elimination C for p = 0, 1, . . . , n/k - 1 and, if n is not a multiple of k, instructs said each node to execute said parallel preprocessing A_1 through A_k, said parallel updating B', and said post-elimination C for p = 0, 1, . . . , [n/k] - 1 and to execute said parallel preprocessing A_1 through $A_{n-[n/k]k}$, said parallel updating B', and said post-elimination C for p = [n/k].

9. Linear calculating equipment comprising:

a network,

a main controller which is connected to said network and executes flow control, and

a plurality of clusters CL_u, where u = 0, 1, . . . , P - 1, each of which is connected to each other by said network and comprises:

a memory which stores blocks of k rows of each coefficient matrix $A^{(r)}$ and corresponding k components of each known vector $b^{(r)}$ and an unknown vector x assigned to said node and expressed by

$$A^{(r)} = (a_{ij}^{(r)}) , \ 1 \leq i, j \leq n,$$

$$b^{(r)} = (b_1^{(r)}, b_2^{(r)}, \ldots, b_n^{(r)})^t,$$

$$x = (x_1, x_2, \ldots, x_n)^t$$

for $0 \leq r \leq n$, for a given system of linear equations

$$A^{(0)}x = b^{(0)},$$

a C gateway which is a junction with the outside of said cluster,

a transmitter which transmits data between said memory and the outside of said cluster through said C gateway, and

a plurality of element processors PE_t, where t = 1 , 2, . . . , P_c, each of which is connected to said memory and comprises:

a pivot choosing section which is connected to said memory, chooses a pivot in the i-th row of $A^{(i-1)}$, and interchanges the i-th column with the chosen pivotal column for i in said element processor's charge,

a preprocessing section A_1 that, immediately after said pivot choosing section determines the transposed pivot

$$a_{pk+1\,pk+1}^{(pk)} \qquad (92)$$

for a given positive integer k and a value of a nonnegative integral parameter p, calculates

$$a_{pk+1\,j}^{(pk+1)} = a_{pk+1\,j}^{(pk)} / a_{pk+1\,pk+1}^{(pk)} \qquad (93)$$

for $pk + 2 \leq j \leq n$ and

$$b_{pk+1}^{(pk+1)} = b_{pk+1}^{(pk)} / a_{pk+1pk+1}^{(pk)} \qquad (94)$$

for pk + 1 in said element processor's charge,

k - 1 preprocessing sections A_t, where t = 2, 3, . . . , k, each of which is connected to said memory and calculates

$$Reg_{pk+t}^{(0)} = a_{pk+tpk+1}^{(pk)}, \qquad (95)$$

$$Reg_{pk+t}^{(1)} = a_{pk+tpk+2}^{(pk)} - Reg_{pk+t}^{(0)} a_{pk+1pk+2}^{(pk+1)}, \qquad (96)$$

. . . ,

$$Reg_{pk+t}^{(t-2)} = a_{pk+tpk+t-1}^{(pk)} - \sum_{m=1}^{t-2} Reg_{pk+t}^{(m-1)} a_{pk+mpk+t-1}^{(pk+m)}, \qquad (97)$$

$$a_{pk+tj}^{(pk+t-1)} = a_{pk+tj}^{(pk)} - \sum_{m=1}^{t-1} Reg_{pk+t}^{(m-1)} a_{pk+mj}^{(pk+m)}, \qquad (98)$$

$$b_{pk+t}^{(pk+t-1)} = b_{pk+t}^{(pk)} - \sum_{m=1}^{t-1} Reg_{pk+t}^{(m-1)} b_{pk+m}^{(pk+m)} \qquad (99)$$

for pk + t ≤ j ≤ n, and, immediately after said pivot choosing section determines the transposed pivot

$$a_{pk+tpk+t}^{(pk+t-1)}, \qquad (100)$$

calculates

$$a_{pk+tj}^{(pk+t)} = a_{pk+tj}^{(pk+t-1)} / a_{pk+tpk+t}^{(pk+t-1)}, \qquad (101)$$

$$b_{pk+t}^{(pk+t)} = b_{pk+t}^{(pk+t-1)} / a_{pk+tpk+t}^{(pk+t-1)} \qquad (102)$$

for pk + t + 1 ≤ j ≤ n,

an updating section B that is connected to said memory, comprises a register set of k registers and an arithmetic unit, and calculates

$$Reg_i^{(0)} = a_{i\,pk+1}^{(pk)}, \tag{103}$$

$$Reg_i^{(1)} = a_{i\,pk+2}^{(pk)} - Reg_i^{(0)}\, a_{pk+1\,pk+2}^{(pk+1)}, \tag{104}$$

. . . . .

$$Reg_i^{(k-1)} = a_{i\,(p+1)k}^{(pk)} - \sum_{m=1}^{k-1} Reg_i^{(m-1)}\, a_{pk+m\,(p+1)k}^{(pk+m)}, \tag{105}$$

$$a_{ij}^{((p+1)k)} = a_{ij}^{(pk)} - \sum_{m=1}^{k} Reg_i^{(m-1)}\, a_{pk+m\,j}^{(pk+m)}, \tag{106}$$

$$b_i^{((p+1)k)} = b_i^{(pk)} - \sum_{m=1}^{k} Reg_i^{(m-1)}\, b_{pk+m}^{(pk+m)} \tag{107}$$

for $(p + 1)k + 1 \leq j \leq n$ for i in said element processor's charge retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in said register set,

a back-substitution section which is connected to said memory and obtains the value of said unknown vector x by calculating

$$x_i = b_i^{(n)} \tag{108}$$

and

$$b_h^{(n+h-i+1)} = b_h^{(n+h-i)} - a_{hi}^{(h)}\, x_i \tag{109}$$

for i and h in said element processor's charge, and

a gateway which is connected to said memory and is a junction with the outside,

10. The equipment as claimed in claim 9, further including:

parallel preprocessing CLA_1 wherein if the (pk + 1)th through (p + 1)k-th rows of said $A^{(0)}$ and corresponding components of said $b^{(0)}$ and x are assigned to said cluster CL_u, then said pivot choosing section of said element processor PE_1 of said CL_u determines the transposed pivot (92) of the (pk + 1)th row, and said preprocessing sections A_1 of said element processors of CL_u simultaneously calculate (93) and (94) for $pk + 2 \leq j \leq n$ and (95) with said each A_1 calculating for elements and components in its charge, and said transmitter transmits the results to said memory of said every other cluster through said C gateway, while said updating section of said element processor in charge of the i-th row calculates (103) for every i such that $(p + 1)k + 1 \leq i \leq n$,

parallel preprocessing CLA_t, where $2 \leq t \leq k$, wherein said preprocessing sections A_t of said cluster

CL_u simultaneously calculate (95) through (99) for $pk + t \leq j \leq n$ with said each A_t calculating for elements and components in its charge, immediately after said pivot choosing section of said PE_t of said CL_u determines the pivot (100), simultaneously calculate (101) and (102) for $pk + t + 1 \leq j \leq n$, and said transmitter transmits the results to said memory of said every other cluster through said C gateway, while said updating section of said element processor in charge of the i-th row calculates

$$Reg_i^{(t-1)} = a_{ipk+t}^{(pk)} - \sum_{m=1}^{k-1} Reg_i^{(m-1)} a_{pk+m\,pk+t}^{(pk+m)} \qquad (110)$$

for every i such that $(p + 1)k + 1 \leq i \leq n$, and

parallel updating means wherein said updating sections B of said each element processor in charge of the i-th row such that $(p + 1)k + 1 \leq i \leq n$ calculate (103) through (107) for $(p + 1)k + 1 \leq j \leq n$ retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in said register set, and back-substitution wherein said back-substitution section of said each cluster in charge of the i-th and h-th components calculates (108) and (109) for $1 \leq h \leq i - 1$ for $i = n, n - 1, \ldots, 1$ in this order of i,.

11. The equipment as claimed in claim 10 wherein said main controller distributes and assigns the rows of said coefficient matrix $A^{(0)}$ and the components of said $b^{(0)}$ and x to said clusters in such a manner as each block of consecutive k rows and corresponding 2k components is transmitted to said memory of said one cluster in the cyclic order of $CL_0, \ldots, CL_{P-1}, CL_0, CL_1, \ldots$, and, if n is a multiple of k, instructs said each cluster to execute said parallel preprocessing CLA_1 through CLA_k and said parallel updating Bc for $p = 0, 1, \ldots, n/k - 2$ and to execute said CLA_1 through CLA_k for $p = n/k - 1$, and, if n is not a multiple of k, instructs said each cluster to execute said CLA_1 through CLA_k and said B_c for $p = 0, 1, \ldots, [n/k] - 1$ and to execute said CLA_1 through $CLA_{n-[n/k]k}$ for $p = [n/k]$, and instructs said each cluster to obtain said unknown vector x by means of said back-substitution.

12. Linear calculating equipment comprising:

a network,

a main controller that is connected to said network and executes flow control, and

a plurality of clusters CL_u, where $u = 0, 1, \ldots, P - 1$, each of which is connected to each other by said network and comprises:

a memory that stores blocks of k rows of each coefficient matrix $A^{(r)}$ and corresponding k components of each known vector $b^{(r)}$ and an unknown vector x assigned to said node and expressed by

$$A^{(r)} = (a_{ij}^{(r)}), \quad 1 \leq i, j \leq n,$$

$$b^{(r)} = (b_1^{(r)}, b_2^{(r)}, \ldots, b_n^{(r)})^t,$$

$$x = (x_1, x_2, \ldots, x_n)^t$$

for $0 \leq r \leq n$, for a given system of linear equations

$$A^{(0)}x = b^{(0)},$$

a C gateway which is a junction with the outside of said cluster,

a transmitter which transmits data between said memory and the outside of said cluster through said C gateway, and

a plurality of element processors PE_t, where t = 1 , 2, . . . , P_c, each of which is connected to said memory and comprises:

a pivot choosing section which is connected to said memory, chooses a pivot in the i-th row of $A^{(i-1)}$, and interchanges the i-th column with the chosen pivotal column for i in said element processor's charge,

a preprocessing section A_1 that, immediately after said pivot choosing section determines the transposed pivot

$$a_{pk+1\,pk+1}^{(pk)} \tag{111}$$

for a given positive integer k and a value of a nonnegative integral parameter p, calculates

$$a_{pk+1\,j}^{(pk+1)} = a_{pk+1\,j}^{(pk)} / a_{pk+1\,pk+1}^{(pk)} \tag{112}$$

for pk + 2 ≤ j ≤ n and

$$b_{pk+1}^{(pk+1)} = b_{pk+1}^{(pk)} / a_{pk+1\,pk+1}^{(pk)} \tag{113}$$

for pk + 1 in said element processor's charge,

k - 1 preprocessing sections A_t, where t = 2, 3, . . . , k, each of which is connected to said memory and calculates

$$Reg_{pk+t}^{(0)} = a_{pk+t\,pk+1}^{(pk)}, \tag{114}$$

$$Reg_{pk+t}^{(1)} = a_{pk+tpk+2}^{(pk)} - Reg_{pk+t}^{(0)} a_{pk+1pk+2}^{(pk+1)}, \quad (115)$$

. . . ,

$$Reg_{pk+t}^{(t-2)} = a_{pk+tpk+t-1}^{(pk)} - \sum_{m=1}^{t-2} Reg_{pk+t}^{(m-1)} a_{pk+mpk+t-1}^{(pk+m)}, \quad (116)$$

$$a_{pk+tj}^{(pk+t-1)} = a_{pk+tj}^{(pk)} - \sum_{m=1}^{t-1} Reg_{pk+t}^{(m-1)} a_{pk+mj}^{(pk+m)}, \quad (117)$$

$$b_{pk+t}^{(pk+t-1)} = b_{pk+t}^{(pk)} - \sum_{m=1}^{t-1} Reg_{pk+t}^{(m-1)} b_{pk+m}^{(pk+m)} \quad (118)$$

for $pk + t \leq j \leq n$, and, immediately after said pivot choosing section determines the transposed pivot

$$a_{pk+tpk+t}^{(pk+t-1)}, \quad (119)$$

calculates

$$a_{pk+tj}^{(pk+t)} = a_{pk+tj}^{(pk+t-1)} / a_{pk+tpk+t}^{(pk+t-1)}, \quad (120)$$

$$b_{pk+t}^{(pk+t)} = b_{pk+t}^{(pk+t-1)} / a_{pk+tpk+t}^{(pk+t-1)} \quad (121)$$

for $pk + t + 1 \leq j \leq n$,

an updating section which is connected to said memory, comprises a register set of k registers and an arithmetic unit, and calculates

$$Reg_i^{(0)} = a_{ipk+1}^{(pk)}, \quad (122)$$

$$Reg_i^{(1)} = a_{i\,pk+2}^{(pk)} - Reg_i^{(0)}\, a_{pk+1\,pk+2}^{(pk+1)}, \tag{123}$$

. . . . .

$$Reg_i^{(k-1)} = a_{i\,(p+1)k}^{(pk)} - \sum_{m=1}^{k-1} Reg_i^{(m-1)}\, a_{pk+m\,(p+1)k}^{(pk+m)}, \tag{124}$$

$$a_{ij}^{((p+1)k)} = a_{ij}^{(pk)} - \sum_{m=1}^{k} Reg_i^{(m-1)}\, a_{pk+m\,j}^{(pk+m)}, \tag{125}$$

$$b_i^{((p+1)k)} = b_i^{(pk)} - \sum_{m=1}^{k} Reg_i^{(m-1)}\, b_{pk+m}^{(pk+m)} \tag{126}$$

in said element processor's charge retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

set,

an updating section which is connected to said pivot choosing section, comprises a register set of k registers and an arithmetic unit, and calculates

$$Reg_i^{(0)} = a_{i\,pk+1}^{(pk)}, \tag{127}$$

$$Reg_i^{(1)} = a_{i\,pk+2}^{(pk)} - Reg_i^{(0)}\, a_{pk+1\,pk+2}^{(pk+1)}, \tag{128}$$

. . . . .

$$Reg_i^{(k-1)} = a_{i(p+1)k}^{(pk)} - \sum_{m=1}^{k-1} Reg_i^{(m-1)} a_{pk+m(p+1)k}^{(pk+m)}, \quad (129)$$

$$a_{ij}^{((p+1)k)} = a_{ij}^{(pk)} - \sum_{m=1}^{k} Reg_i^{(m-1)} a_{pk+mj}^{(pk+m)}, \quad (130)$$

$$b_i^{((p+1)k)} = b_i^{(pk)} - \sum_{m=1}^{k} Reg_i^{(m-1)} b_{pk+m}^{(pk+m)} \quad (131)$$

for $(p + 1)k + 1 \leq j \leq n$ retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in said register set,

k - 1 postprocessing sections C_t, where t = 1, 2, . . . , k - 1, each of which is connected to said pivot choosing section and calculates

$$Reg^{(0)} = a_{pk+1pk+t+1}^{(pk+t)}, \quad (132)$$

$$Reg^{(1)} = a_{pk+2pk+t+1}^{(pk+t)}, \quad (133)$$

. . . ,

$$Reg^{(t-1)} = a_{pk+tpk+t+1}^{(pk+t)}, \quad (134)$$

$$a_{pk+1j}^{(pk+t+1)} = a_{pk+1j}^{(pk+t)} - Reg^{(0)} a_{pk+t+1j}^{(pk+t+1)}, \quad (135)$$

$$a_{pk+2j}^{(pk+t+1)} = a_{pk+2j}^{(pk+t)} - Reg^{(1)}\, a_{pk+t+1j}^{(pk+t+1)}, \qquad (136)$$

. . . ,

$$a_{pk+tj}^{(pk+t+1)} = a_{pk+tj}^{(pk+t)} - Reg^{(t-1)}\, a_{pk+t+1j}^{(pk+t+1)}, \qquad (137)$$

$$b_{pk+1}^{(pk+t+1)} = b_{pk+1}^{(pk+t)} - Reg^{(0)}\, b_{pk+t+1}^{(pk+t+1)}, \qquad (138)$$

$$b_{pk+2}^{(pk+t+1)} = b_{pk+2}^{(pk+t)} - Reg^{(1)}\, b_{pk+t+1}^{(pk+t+1)}, \qquad (139)$$

. . . . .

$$b_{pk+t}^{(pk+t+1)} = b_{pk+t}^{(pk+t)} - Reg^{(t-1)}\, b_{pk+t+1}^{(pk+t+1)} \qquad (140)$$

$$Reg^{(0)} = a_{pk+1pk+t+1}^{(pk+t)}, \qquad (141)$$

$$Reg^{(1)} = a_{pk+2pk+t+1}^{(pk+t)}, \qquad (142)$$

. . . ,

$$Reg^{(t-1)} = a_{pk+tpk+t+1}^{(pk+t)}, \qquad (143)$$

$$a_{pk+1j}^{(pk+t+1)} = a_{pk+1j}^{(pk+t)} - Reg^{(0)}\, a_{pk+t+1j}^{(pk+t+1)}, \qquad (144)$$

$$a_{pk+2j}^{(pk+t+1)} = a_{pk+2j}^{(pk+t)} - Reg^{(1)}\, a_{pk+t+1j}^{(pk+t+1)}, \qquad (145)$$

. . . ,

$$a_{pk+tj}^{(pk+t+1)} = a_{pk+tj}^{(pk+t)} - Reg^{(t-1)}\, a_{pk+t+1j}^{(pk+t+1)}, \qquad (146)$$

$$b_{pk+1}^{(pk+t+1)} = b_{pk+1}^{(pk+t)} - Reg^{(0)}\, b_{pk+t+1}^{(pk+t+1)}, \qquad (147)$$

$$b_{pk+2}^{(pk+t+1)} = b_{pk+2}^{(pk+t)} - Reg^{(1)}\, b_{pk+t+1}^{(pk+t+1)}, \qquad (148)$$

. . . . .

$$b_{pk+t}^{(pk+t+1)} = b_{pk+t}^{(pk+t)} - Reg^{(t-1)}\, b_{pk+t+1}^{(pk+t+1)} \qquad (149)$$

for $pk + t + 2 \leq j \leq n$, and

a gateway which is connected to said pivot choosing section and is a junction with the outside of said element processor.

13. The equipment as claimed in claim 12, further including:

parallel preprocessing CLA_1 wherein if the $(pk + 1)$th through $(p + 1)k$-th rows of said $A^{(0)}$ and corresponding components of said $b^{(0)}$ and x are assigned to said cluster CL_u, then said pivot choosing section of said element processor PE_1 of said CL_u determines the transposed pivot (111) of the $(pk + 1)$th row, and said preprocessing sections A_1 of said element processors of said CL_u simultaneously calculate (112) and (113) for $pk + 2 \leq j \leq n$ with said each A_1 calculating for elements and components in its charge, and said transmitter transmits the results to said memory of said every other cluster through said C gateway, while said updating section B' of said element processor in charge of the i-th row calculates (122) for every i such that $(p + 1)k + 1 \leq i \leq n$,

parallel preprocessing A_t, where $t = 2, 3, \ldots, k$, wherein said preprocessing sections A_t of said element processors of said cluster CL_u simultaneously calculate (114) through (118) for $pk + t \leq j \leq n$ with said each A_t calculating for elements and components in its charge and, immediately after said pivot choosing section of said PE_t of said CL_u determines the pivot (119), simultaneously calculate (120) and (121) for $pk + t + 1 \leq j \leq n$, and said transmitter transmits the results to said memory of said every other cluster through said C gateway, while said updating section B' of said element processor in charge of the i-th row calculates

$$Reg_i^{(t-1)} = a_{ipk+t}^{(pk)} - \sum_{m=1}^{k-1} Reg_i^{(m-1)} a_{pk+mpk+t}^{(pk+m)} \qquad (150)$$

for every i such that $(p + 1)k + 1 \leq i \leq n$,

parallel updating B'_c wherein said updating section B' of said each element processor in charge of the i-th row such that $1 \leq i \leq pk$ or $(p + 1)k + 1 \leq i \leq n$ if n is a multiple of k or $p < [n/k]$ and $1 \leq i \leq -[n/k]k$ otherwise also calculates (122) through (126) for $(p + 1)k + 1 \leq j \leq n$ if n is a multiple of k or $p < [n/k]$ and for $[n/k]k + 1 \leq j \leq n$ otherwise, retaining the values of

$$Reg_i^{(0)}, \ldots, Reg_i^{(k)}$$

in the register set, and

post-elimination C_c wherein said postprocessing sections C_t of said element processors of said CL_u simultaneously calculate (132) through (140) for j such that $pk + t + 2 \leq j \leq n$ for $t = 1, 2, \ldots, k - 1$ if n is a multiple of k or $p < [n/k]$ and for $t = 1, 2, \ldots, n - [n/k]k$ otherwise.

14. The equipment as claimed in claim 13 wherein said main controller distributes and assigns the rows of said coefficient matrix $A^{(0)}$ and the components of said $b^{(0)}$ and x to said clusters in such a manner as each block of consecutive k rows and corresponding 2k components is transmitted to said memory of said one cluster in the cyclic order of $CL_0, \ldots, CL_{P-1}, CL_0, CL_1, \ldots$, and, if n is a multiple of k, instructs said each cluster to execute said parallel preprocessing CLA_1 through CLA_k, said parallel updating B'_c and said parallel postelimination C_c for $p = 0, 1, \ldots, n/k - 1$, and if n is not a multiple of k, instructs said each cluster to execute said parallel preprocessing CLA_1 through CLA_k, said parallel updating B'_c, and said postelimination C_c for $p = 0, 1, \ldots, [n/k] - 1$ and to execute said parallel preprocessing CLA_1 through $CLA_{n-[n/k]k}$, said parallel updating B'_c, and said postelimination C_c for $p = [n/k]$.

15. A parallel computer composed of C clusters $CL_1, \ldots, CL_C$ connected by a network, said each cluster comprising P_c element processors and a shared memory that stores part of each coefficient matrix $A^{(r)}$ and each known vector $b^{(r)}$ and an unknown vector x expressed by

$$A^{(r)} = (a_{ij}^{(r)}),\ 1 \le i, j \le n,$$

$$b^{(r)} = (b_1^{(r)}, b_2^{(r)}, \ldots, b_n^{(r)})^t,$$

$x = (x_1, x_2, \ldots, x_n)^t$

for $0 \le r \le n$, for a given system of linear equations

$0A^{(0)}x = b^{(0)}$,

said parallel computer comprising:

a data distribution means which distributes the rows of said coefficient matrix $A^{(0)}$ and the components of said vectors $b^{(0)}$ and x to said shared memory of said clusters in such a manner as each block of consecutive k rows and corresponding 2k components is transmitted to said shared memory in the cyclic order of $CL_1, \ldots, CL_C, CL_1, CL_2, \ldots$, and assigns those distributed to said cluster's shared memory to its said element processors row by row,

a pivot choosing means which chooses a pivot in a row assigned to said each element processor,

an elementary pre-elimination means which, after said pivot choosing means chooses the pivot

$$a_{kP_c+1\,kP_c+1}^{(kP_c)},$$

calculates

$$a_{kP_c+1\,j}^{(kP_c+1)} = a_{kP_c+1\,j}^{(kP_c)} / a_{kP_c+1\,kP_c+1}^{(kP_c)},$$

$$b_{kP_c+1}^{(kP_c+1)} = b_{kP_c+1}^{(kP_c)} / a_{kP_c+1\,kP_c+1}^{(kP_c)}$$

in said element processor in charge of the $(kP_c + 1)$th row, transmits the results to said shared memory of said every other cluster to which said element processor in charge of an i-th row such that $kP_c + 1 \le i \le n$ belongs, and, for $l = 2, \ldots, P_c$, calculates

$$t_i^{(l-1)} = a_{i\,kP_c+l}^{(kP_c)} - a_{i\,kP_c+1}^{(kP_c)} a_{kP_c+1\,kP_c+l}^{(kP_c+1)} - \sum_{m=2}^{l-1} t_i^{(m-1)} a_{kP_c+m\,kP_c+l}^{(kP_c+m)}$$

for $kP_c + l \le i \le n$ in said element processor in charge of the i-th row, calculates

$$a_{kP_c+1j}^{(kP_c+l-1)} = a_{kP_c+1j}^{(kP_c)} - a_{kP_c+1kP_c+1}^{(kP_c)} a_{kP_c+1j}^{(kP_c+1)} - \sum_{m=2}^{l-1} t_{kP_c+1}^{(m-1)} a_{kP_c+mj}^{(kP_c+m)},$$

$$b_{kP_c+1}^{(kP_c+l-1)} = b_{kP_c+1}^{(kP_c)} - a_{kP_c+1kP_c+1}^{(kP_c)} b_{kP_c+1}^{(kP_c+1)} - \sum_{m=2}^{l-1} t_{kP_c+1}^{(m-1)} b_{kP_c+m}^{(kP_c+m)}$$

in said element processor in charge of the (kP_c + l)th row, and, after said pivot choosing means determines the pivot

$$a_{kP_c+1kP_c+1}^{(kP_c+l-1)},$$

calculates

$$a_{kP_c+1j}^{(kP_c+l)} = a_{kP_c+1j}^{(kP_c+l-1)} / a_{kP_c+1kP_c+1}^{(kP_c+l-1)},$$

$$b_{kP_c+1}^{(kP_c+l)} = b_{kP_c+1}^{(kP_c+l-1)} / a_{kP_c+1kP_c+1}^{(kP_c+l-1)}$$

in said element processor in charge of the (kP_c + l)th row, transmits the results

$$a_{kP_c+1j}^{(kP_c+l)} = a_{kP_c+1j}^{(kP_c+l-1)} / a_{kP_c+1kP_c+1}^{(kP_c+l-1)},$$

$$b_{kP_c+1}^{(kP_c+l)} = b_{kP_c+1}^{(kP_c+l-1)} / a_{kP_c+1kP_c+1}^{(kP_c+l-1)}$$

to said shared memory of every other cluster to which said element processor in charge of an i-th row such that $kP_c + l + 1 \leq i \leq n$ belongs,

a multi-pivot elimination means which calculates

$$a_{ij}^{((k+1)P_c)} = a_{ij}^{(kP_c)} - a_{ikP_c+1}^{(kP_c)} a_{kP_c+1j}^{(kP_c+1)} - \sum_{m=2}^{P_c} t_i^{(m-1)} a_{kP_c+mj}^{(kP_c+m)},$$

$$b_i^{((k+1)P_c)} = b_i^{(kP_c)} - a_{ikP_c+1}^{(kP_c)} b_{kP_c+1}^{(kP_c+1)} - \sum_{m=2}^{P_c} t_i^{(m-1)} b_{kP_c+m}^{(kP_c+m)}$$

in said each element processor in charge of the i-th row such that $(k + 1)P_c + 1 \leq i \leq n$,

a means for testing if the operation of said multi-pivot elimination means was repeated $[n/P_c]$ times, and

a remainder elimination means that executes said elementary pre-elimination means for the $([n/P_c]-P_c + 1)$th row through the n-th row, if said testing means judges that the operation of said multi-pivot elimination means was executed $[n/P_c]$ times, and n is not a multiple of P_c.

16. The parallel computer as claimed in claim 15, further including

an elementary back-substitution means that calculates

$$x_i = b_i^{(n)}$$

in said element processor in charge of the i-th row after the elimination process of said parallel elimination method,

an elementary back-transmission means that transmits x_i to said shared memory of said every cluster to which said element processor in charge of an h-th row such that $1 \leq h \leq i - 1$ belongs,

an elementary back-calculation means which calculates

$$b_h^{(n+h-i+1)} = b_h^{(n+h-i)} - a_{hi}^{(h)} x_i$$

for $1 \leq h \leq i - 1$ in said element processor in charge of the h-th row, and

a means for testing if the operation of said elementary back-substitution means was repeated from $i = n$ to $i = 1$.

17. The parallel computer as claimed in claim 15 wherein a pivot choosing method comprising:

a search means whereby said element processor searches for a nonzero element in the order of increasing column numbers from the diagonal element in the same row, if a diagonal element of said coefficient matrix $A^{(r)}$ is 0,

a column number broadcasting means that notifies said other element processors of the column number of a nonzero element found by said search means,

an element interchange means whereby said each element processor interchanges the two elements which are in its charge and have the same column numbers as the above diagonal zero element and the found nonzero element, and

a component interchange means whereby said two element processors interchange the two components of said unknown vector which are in their charge and have the same component indices as the column numbers of said diagonal zero element and said found nonzero element.

18. The parallel computer as claimed in claim 15 wherein said pivot choosing method comprising:

a search means whereby said element processor searches for an element with the greatest absolute value in the order of increasing column numbers from a diagonal element in the same row,

a column number broadcasting means that notifies said other element processors of the column number of an element found by said search means,

an element interchange means whereby said each element processor interchanges the two elements which are in its charge and have the same column number as said diagonal element and the found element, and

a component interchange means whereby said two element processors interchange the two components of the unknown vector which are in their charge and have the same component indices as the column numbers of said diagonal element and said found component.

19. A parallel computer composed of C clusters $CL_1, \ldots, CL_C$ connected by a network, said each cluster comprising P_c element processors and a shared memory which stores part of each coefficient matrix $A^{(r)}$ and each known vector $b^{(r)}$ and an unknown vector x expressed by

$$A^{(r)} = (a_{ij}^{(r)}) , \ 1 \leq i, j \leq n,$$

$$b^{(r)} = (b_1^{(r)}, b_2^{(r)}, \ldots, b_n^{(r)})^t,$$

$x = (x_1, x_2, \ldots, x_n)^t$

for $0 \leq r \leq n$, for a given system of linear equations

$A^{(0)}x = b^{(0)}$,

said parallel computer comprising:

a data distribution means which distributes the rows of said coefficient matrix $A^{(0)}$ and the components of said vectors $b^{(0)}$ and x to said shared memory of said clusters in such a manner as each block of consecutive k rows and corresponding 2k components is transmitted to said shared memory in the cyclic order of $CL_1, \ldots, CL_C, CL_1, CL_2, \ldots$, and assigns those distributed to said cluster's shared memory to its said element processors row by row,

a pivot choosing means which chooses a pivot in a row assigned to said each element processor,

an elementary pre-elimination means which, after said pivot choosing means chooses the pivot

$$a_{kP_c+1\,kP_c+1}^{(kP_c)},$$

calculates

$$a_{kP_c+1\,j}^{(kP_c+1)} = a_{kP_c+1\,j}^{(kP_c)} / a_{kP_c+1\,kP_c+1}^{(kP_c)},$$

$$b_{kP_c+1}^{(kP_c+1)} = b_{kP_c+1}^{(kP_c)} / a_{kP_c+1\,kP_c+1}^{(kP_c)}$$

in said element processor in charge of the $(kP_c + 1)$th row, transmits the results to said shared memory of said every other cluster to which said element processor in charge of an i-th row such that $kP_c + 1 \leq i \leq n$ belongs, and, for $l = 2, \ldots, P_c$, calculates

$$t_i^{(l-1)} = a_{i\,kP_c+l}^{(kP_c)} - a_{i\,kP_c+1}^{(kP_c)} a_{kP_c+1\,kP_c+l}^{(kP_c+1)} - \sum_{m=2}^{l-1} t_i^{(m-1)} a_{kP_c+m\,kP_c+l}^{(kP_c+m)}$$

for $kP_c + l \leq i \leq n$ in said element processor in charge of the i-th row, calculates

$$a_{kP_c+lj}^{(kP_c+l-1)} = a_{kP_c+lj}^{(kP_c)} - a_{kP_c+lkP_c+1}^{(kP_c)} a_{kP_c+1j}^{(kP_c+1)} - \sum_{m=2}^{l-1} t_{kP_c+l}^{(m-1)} a_{kP_c+mj}^{(kP_c+m)},$$

$$b_{kP_c+l}^{(kP_c+l-1)} = b_{kP_c+l}^{(kP_c)} - a_{kP_c+lkP_c+1}^{(kP_c)} b_{kP_c+1}^{(kP_c+1)} - \sum_{m=2}^{l-1} t_{kP_c+l}^{(m-1)} b_{kP_c+m}^{(kP_c+m)}$$

in said element processor in charge of the $(kP_c + l)$th row, and, after said pivot choosing means determines the pivot

$$a_{kP_c+lkP_c+l}^{(kP_c+l-1)},$$

calculates

$$a_{kP_c+1j}^{(kP_c+1)} = a_{kP_c+1j}^{(kP_c+l-1)} / a_{kP_c+lkP_c+l}^{(kP_c+l-1)},$$

$$b_{kP_c+1}^{(kP_c+1)} = b_{kP_c+1}^{(kP_c+l-1)} / a_{kP_c+lkP_c+l}^{(kP_c+l-1)}$$

in said element processor in charge of the $(kP_c + l)$th row, transmits the results

$$a_{kP_c+1j}^{(kP_c+1)} = a_{kP_c+1j}^{(kP_c+l-1)} / a_{kP_c+lkP_c+l}^{(kP_c+l-1)},$$

$$b_{kP_c+1}^{(kP_c+1)} = b_{kP_c+1}^{(kP_c+l-1)} / a_{kP_c+lkP_c+l}^{(kP_c+l-1)}$$

to said shared memory of every other cluster to which said element processor in charge of an i-th row such that $kP_c + l + 1 \leq i \leq n$ belongs,

a multi-pivot elimination means that calculates

$$a_{ij}^{((k+1)P_c)} = a_{ij}^{(kP_c)} - a_{ikP_c+1}^{(kP_c)} a_{kP_c+1j}^{(kP_c+1)} - \sum_{m=2}^{P_c} t_i^{(m-1)} a_{kP_c+mj}^{(kP_c+m)},$$

$$b_i^{((k+1)P_c)} = b_i^{(kP_c)} - a_{ikP_c+1}^{(kP_c)} b_{kP_c+1}^{(kP_c+1)} - \sum_{m=2}^{P_c} t_i^{(m-1)} b_{kP_c+m}^{(kP_c+m)}$$

in said each element processor in charge of the i-th row such that $(k + 1)P_c + 1 \leq i \leq n$,

an elementary post-elimination means which calculates

$$a_{ij}^{(r+1)} = a_{ij}^{(r)} - a_{il+1}^{(r)} a_{i+1j}^{(r+1)},$$

$$b_i^{(r+1)} = b_i^{(r)} - a_{il+1}^{(r)} b_{i+1}^{(r+1)}$$

in said element processor in charge of the i-th row,

a post-elimination processing means which calculates

$$a_{ij}^{(r+1)} = a_{ij}^{(r)} - a_{il+1}^{(r)} a_{i+1j}^{(r+1)},$$

$$b_i^{(r+1)} = b_i^{(r)} - a_{il+1}^{(r)} b_{i+1}^{(r+1)}$$

for $l = -w + q + 1$ for $w = 1, \ldots, q$ and $q = 1, \ldots, P_c - 1$ for $kP_c + 1 \leq i \leq kP_c + q$ in said element processor in charge of the i-th row,

a means for testing if the operation of said post-elimination means was executed $[n/P_c]$ times, and

a remainder elimination means that executes said elementary pre-elimination means for the $([n/P_c]-P_c + 1)$th through the n-th rows and executes said multi-pivot elimination means and said post-elimination means, if the testing means judges that the operation of said post-elimination means was executed $[n/P_c]$ times.

20. The parallel computer as claimed in claim 18 wherein a pivot choosing method comprising:

a search means whereby said element processor searches for a nonzero element in the order of increasing column numbers from the diagonal element in the same row, if a diagonal element of said coefficient matrix $A^{(r)}$ is 0,

a column number broadcasting means that notifies said other element processors of the column number of a nonzero element found by said search means,

an element interchange means whereby said each element processor interchanges the two elements which are in its charge and have the same column numbers as the above diagonal zero element and the found nonzero element, and

a component interchange means whereby said two element processors interchange the two components of said unknown vector which are in their charge and have the same component indices as the column numbers of said diagonal zero element and said found nonzero element.

21. The parallel computer claim 18 wherein said pivot choosing method comprising:

a search means whereby said element processor searches for an element with the greatest absolute value in the order of increasing column numbers from a diagonal element in the same row,

a column number broadcasting means that notifies said other element processors of the column number of an element found by said search means,

an element interchange means whereby said each element processor interchanges the two elements which are in its charge and have the same column number as said diagonal element and the found element, and

a component interchange means whereby said two element processors interchange the two components of the unknown vector which are in their charge and have the same component indices as the column numbers of said diagonal element and said found component.

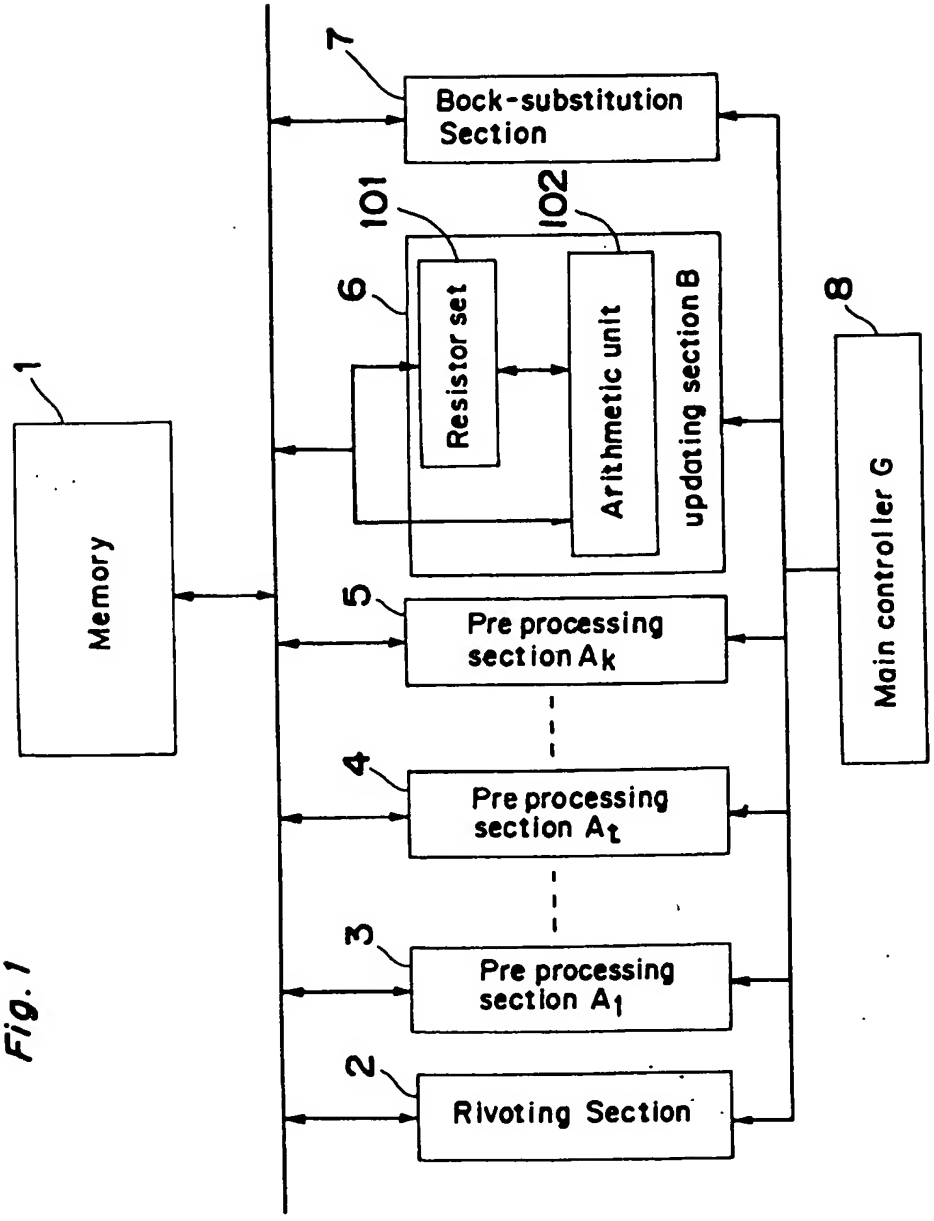
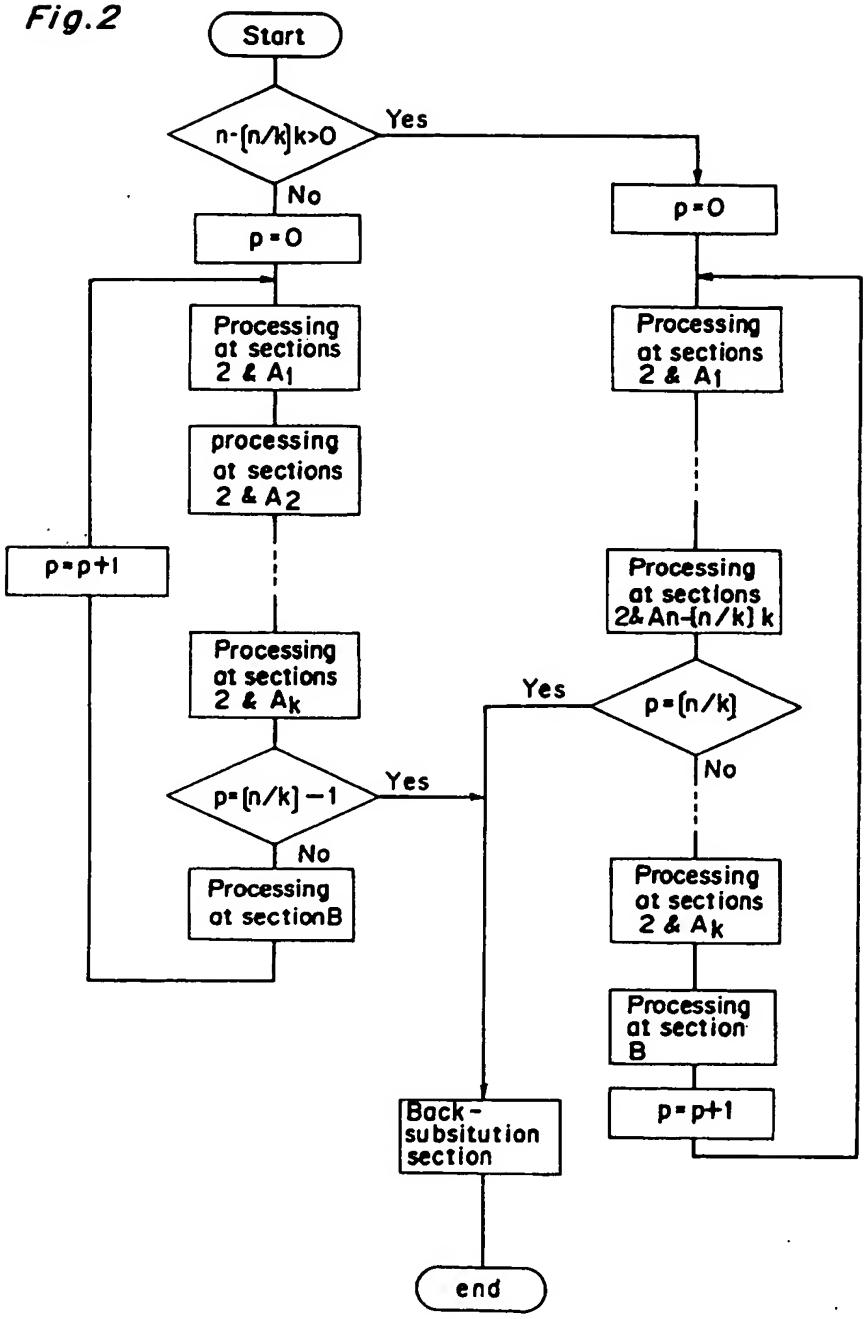

7
Bock-substitution Section
101
102
6
Resistor set
Arithmetic unit
updating section B
8
1
Memory
Main controller G
5
Pre processing section A_k
4
Pre processing section A_t
3
Pre processing section A_1
2
Rivoting Section

Fig. 1

Fig.2


Start
n-[n/k]k>0
Yes
No
p=0
Processing at sections 2 & A1
processing at sections 2 & A2
p=p+1
Processing at sections 2 & Ak
p=[n/k]-1
Yes
No
Processing at sectionB
p=0
Processing at sections 2 & A1
Processing at sections 2& An-[n/k] k
p=[n/k]
Yes
No
Processing at sections 2 & Ak
Processing at section B
p=p+1
Back-subsitution section
end

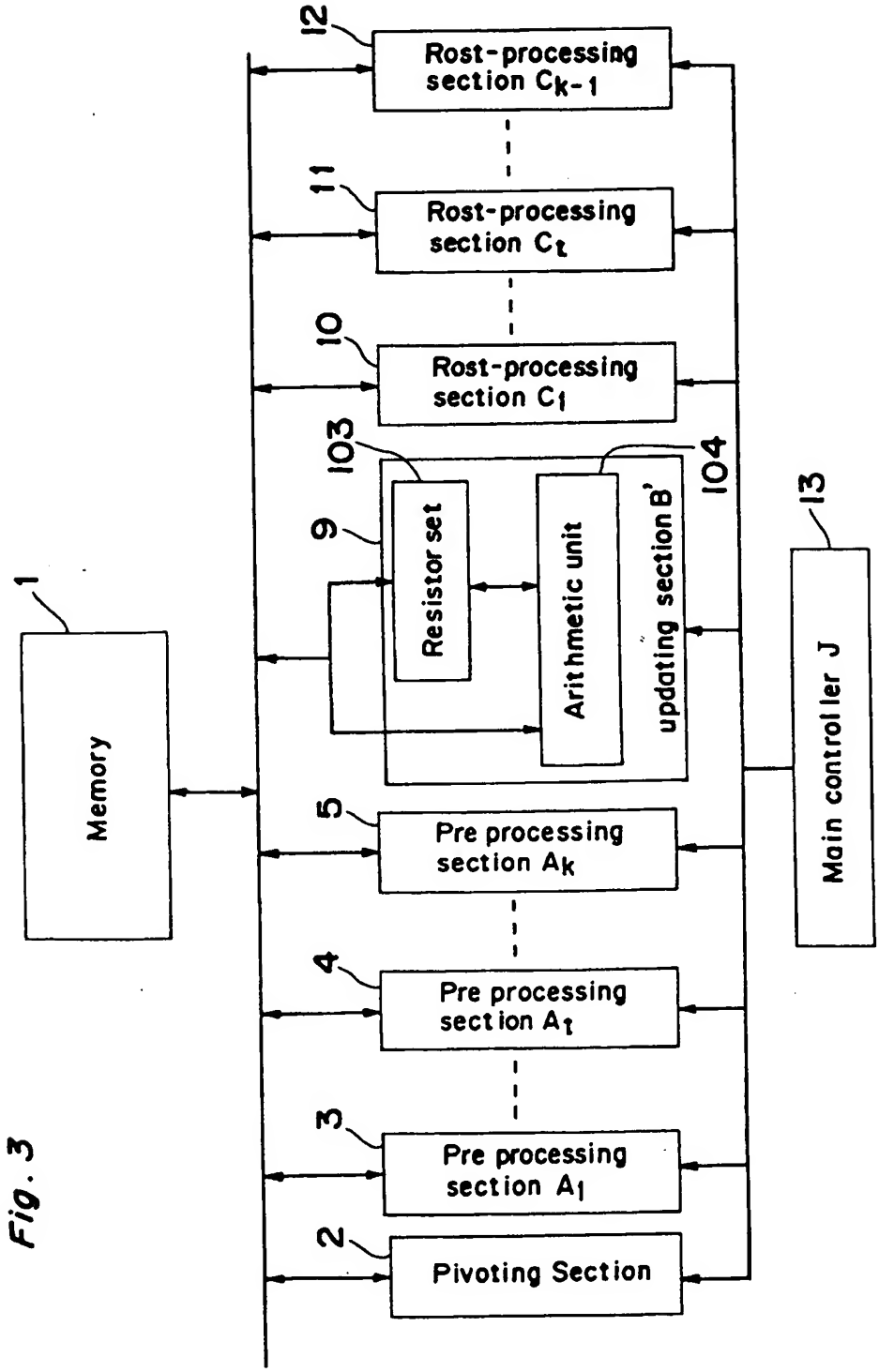
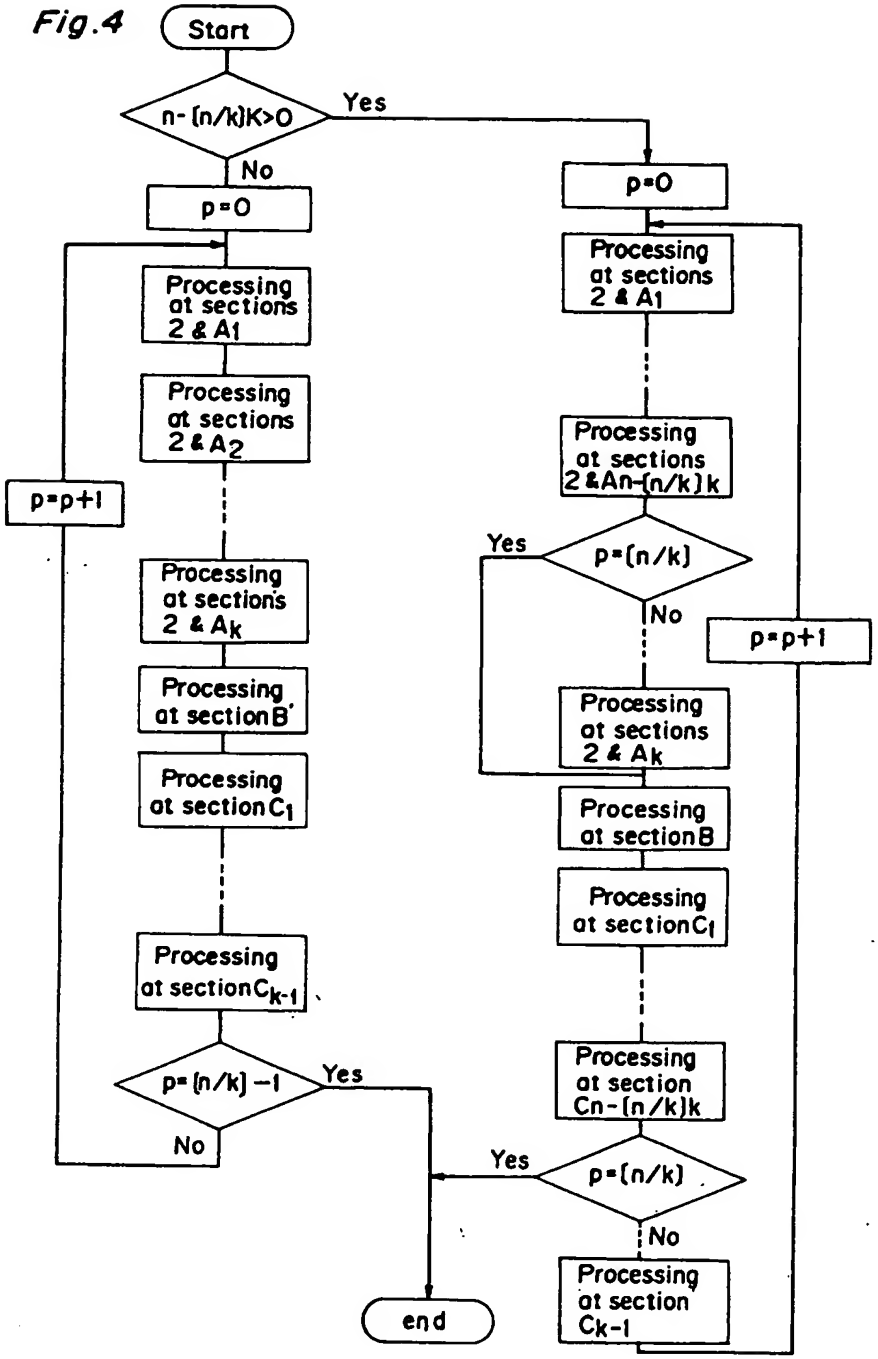

12
Rost-processing section C_{k-1}
11
Rost-processing section C_t
10
Rost-processing section C_1
103
104
9
Resistor set
Arithmetic unit
updating section B'
13
1
Memory
Main controller J
5
Pre processing section A_k
4
Pre processing section A_t
3
Pre processing section A_1
2
Pivoting Section

Fig. 3

Fig.4


Start
n-[n/k]K>0
Yes
No
p=0
Processing at sections 2 & A1
Processing at sections 2 & A2
p=p+1
Processing at sections 2 & Ak
Processing at section B
Processing at section C1
Processing at section Ck-1
p=[n/k]-1
Yes
No
end
p=0
Processing at sections 2 & A1
Processing at sections 2 & An-[n/k]k
Yes
p=[n/k]
No
p=p+1
Processing at sections 2 & Ak
Processing at section B
Processing at section C1
Processing at section Cn-[n/k]k
Yes
p=[n/k]
No
Processing at section Ck-1

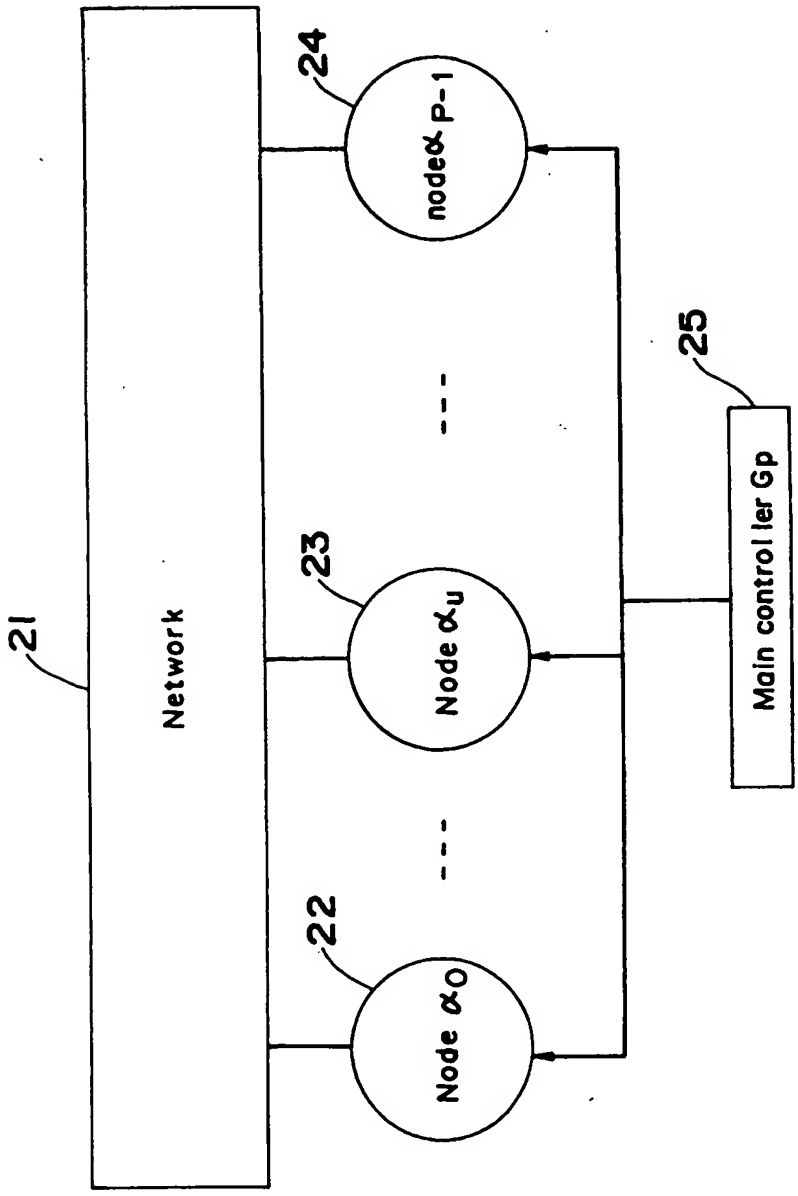

21
Network
22
Node α_0
23
Node α_u
24
node α_{P-1}
25
Main controller Gp

Fig.5

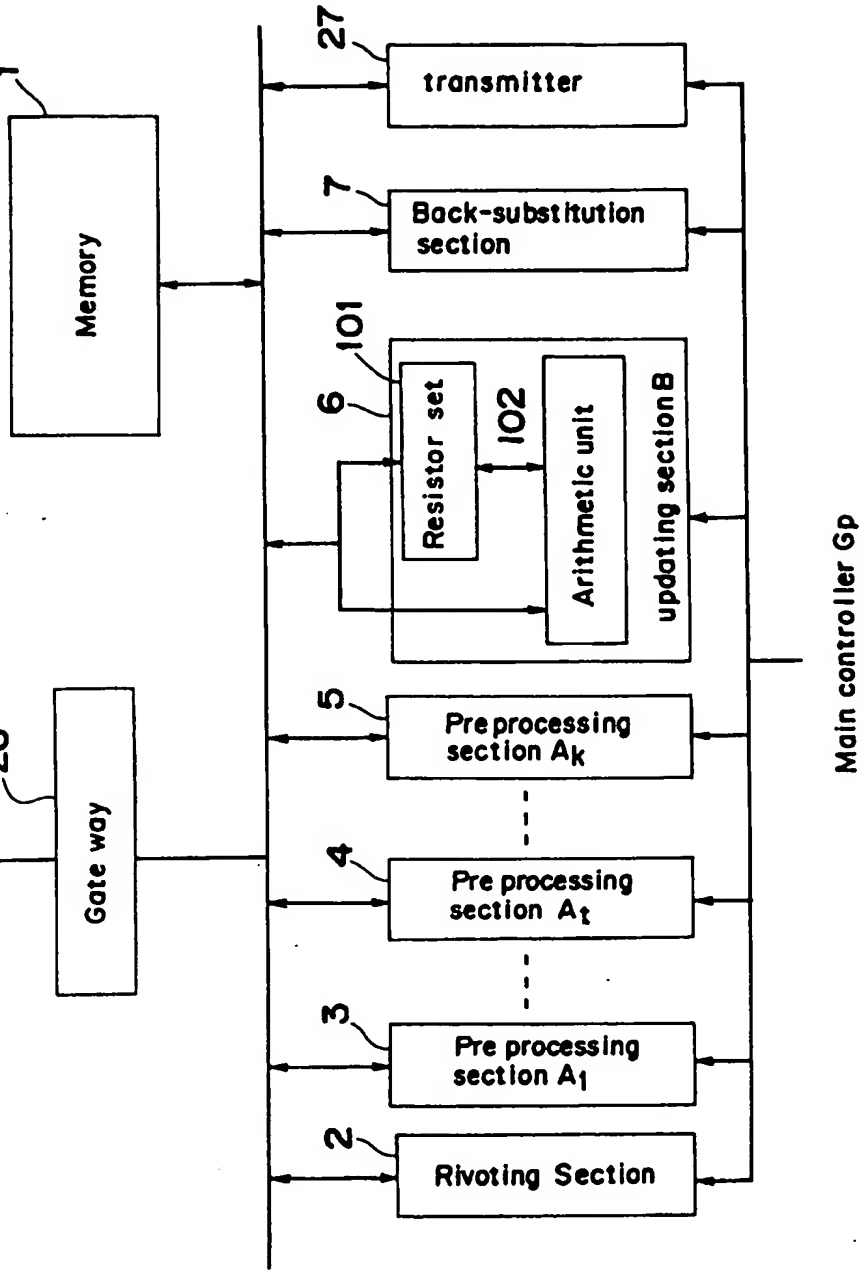
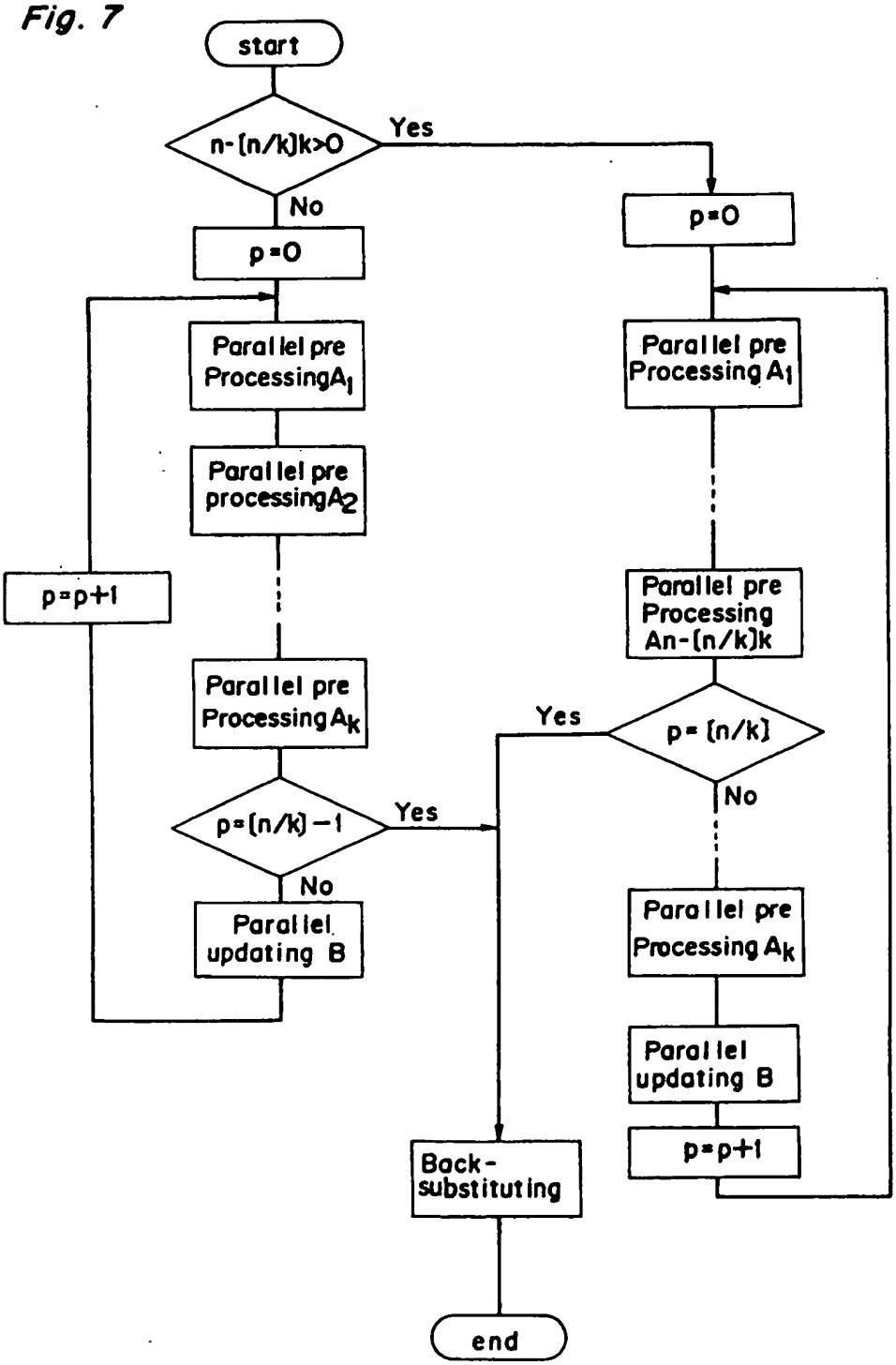

1
Memory
27
transmitter
7
Back-substitution section
101
6
Resistor set
102
Arithmetic unit
updating section B
5
Preprocessing section A_k
4
Pre processing section A_t
3
Pre processing section A_1
2
Rivoting Section
26
Gate way
Main controller Gp

Fig.6

Fig. 7


start
n-[n/k]k>0
Yes
No
p=0
p=0
Parallel pre Processing A_1
Parallel pre processing A_2
p=p+1
Parallel pre Processing A_k
p=[n/k]-1
Yes
No
Parallel updating B
Parallel pre Processing A_1
Parallel pre Processing An-[n/k]k
Yes
p=[n/k]
No
Parallel pre Processing A_k
Parallel updating B
p=p+1
Back-substituting
end

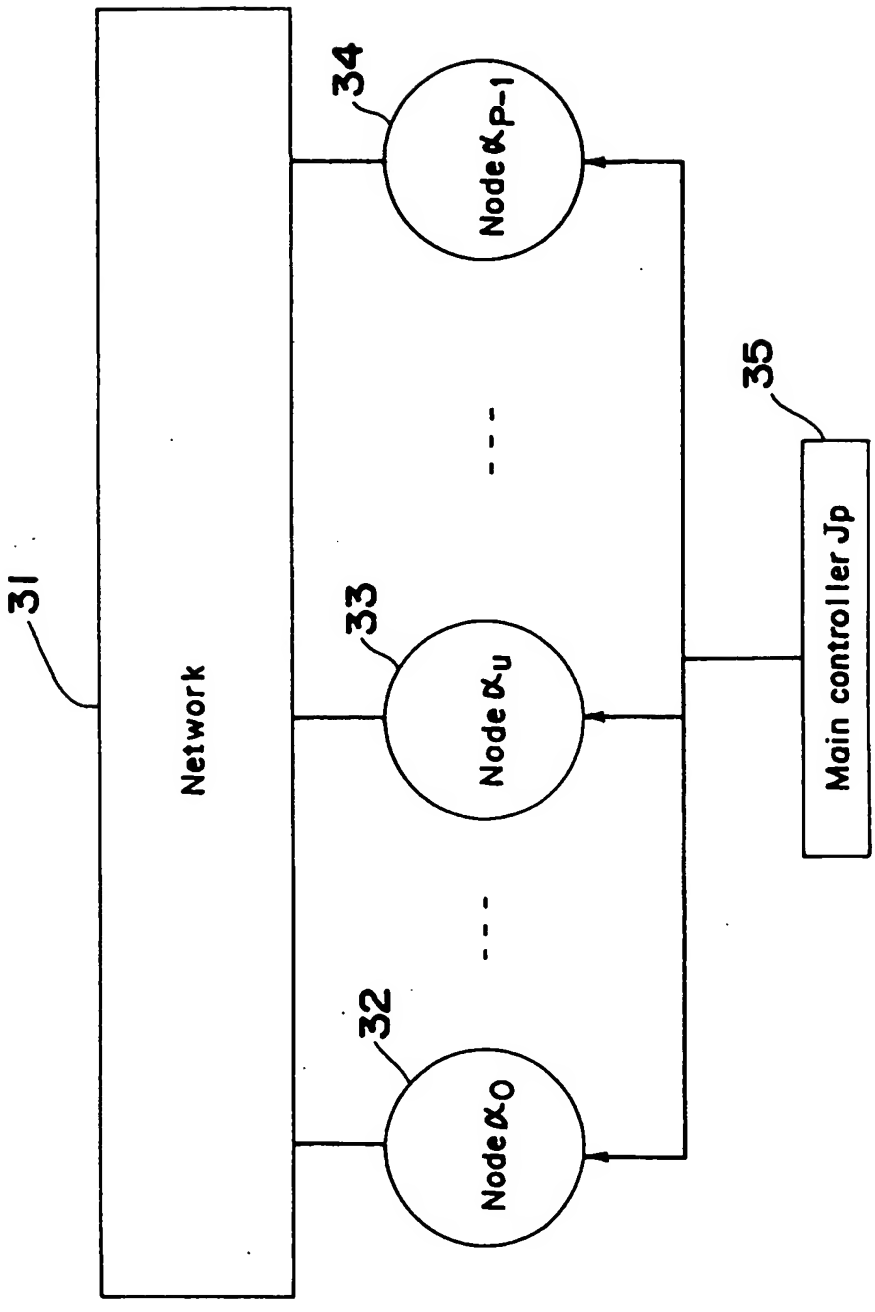

31
Network
34
Node α_{P-1}
33
Node α_u
32
Node α_0
35
Main controller Jp

Fig. 8

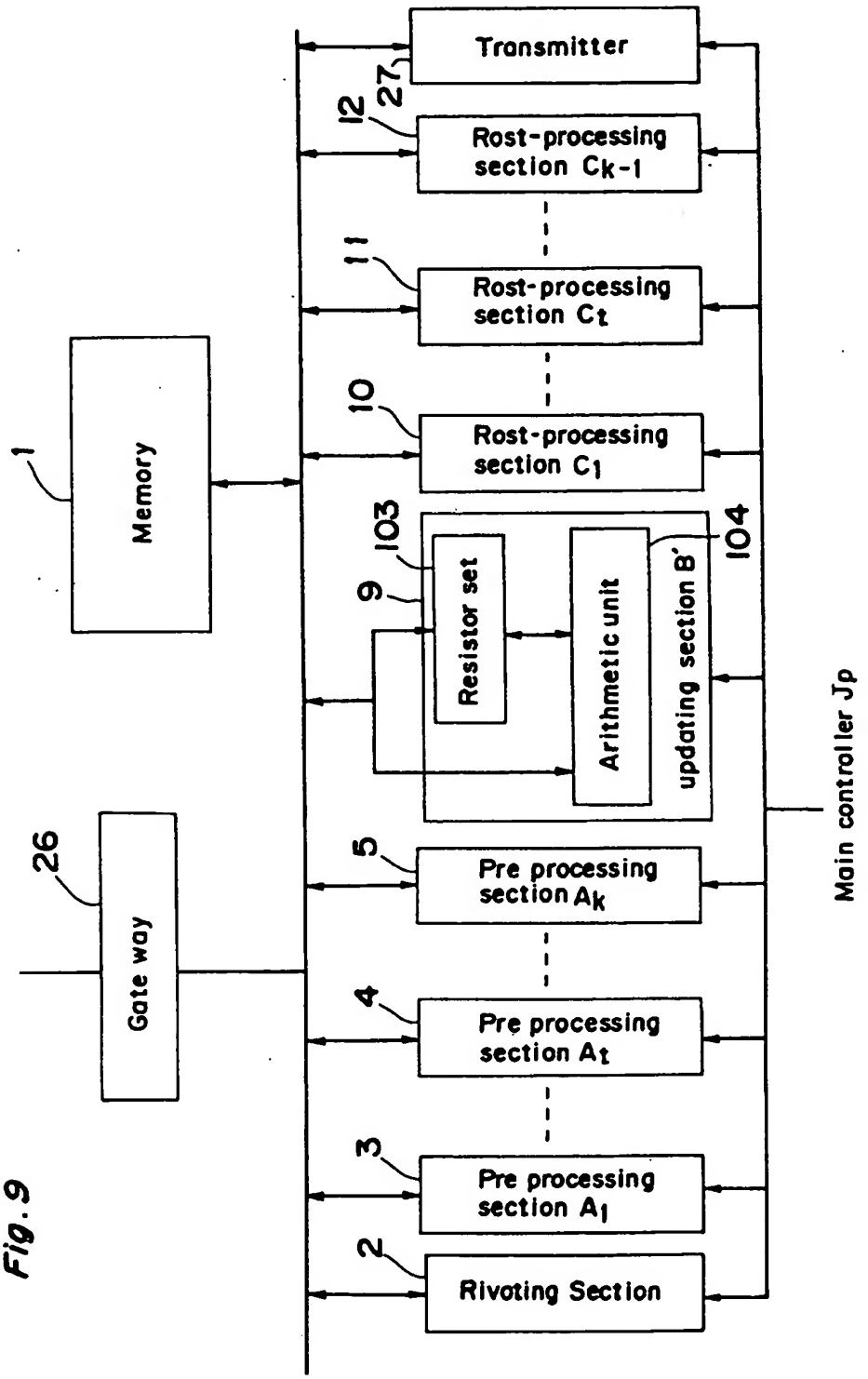
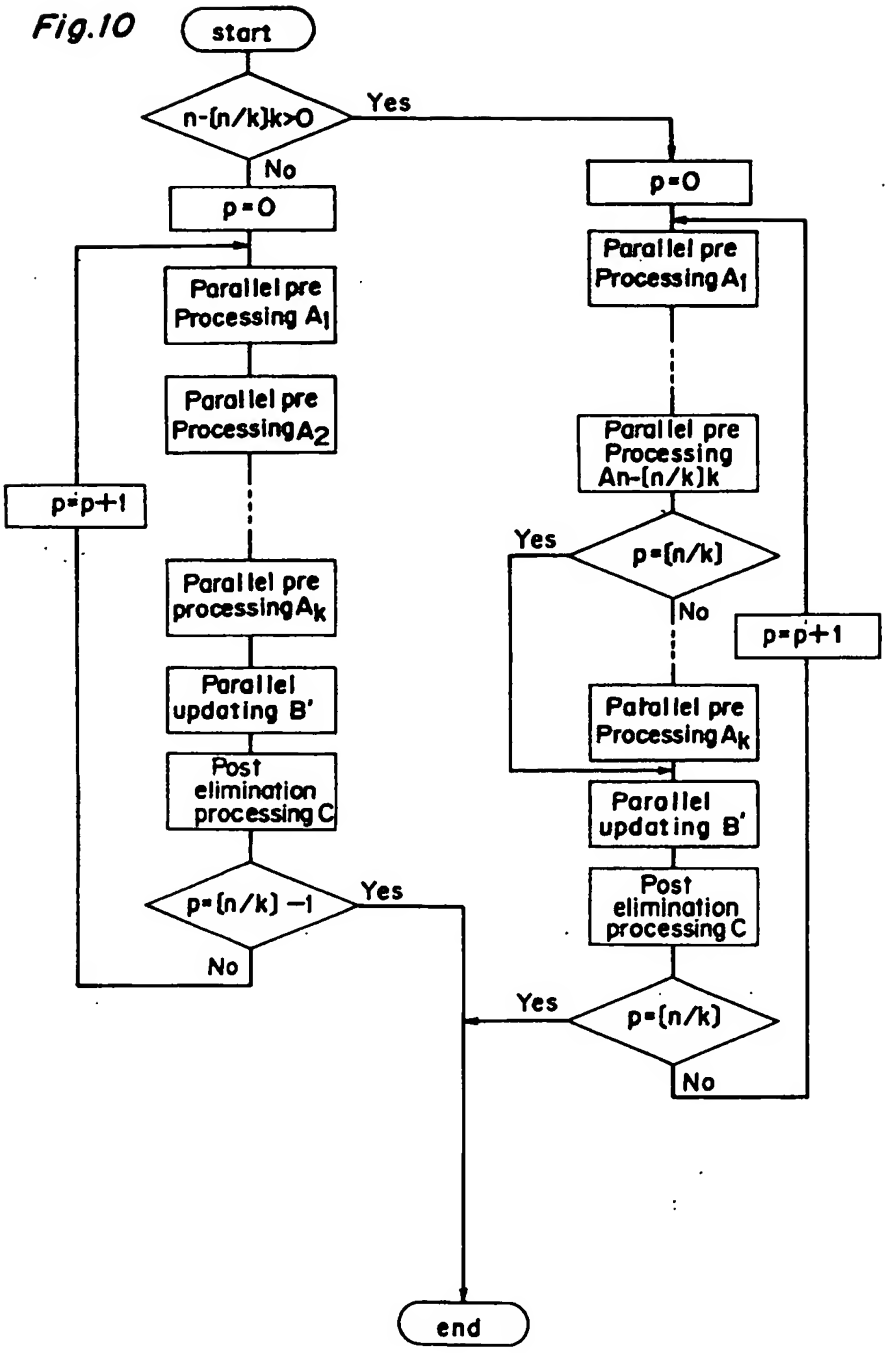

Transmitter
27
12
Rost-processing section Ck-1
11
Rost-processing section Ct
10
Rost-processing section C1
1
Memory
103
9
Resistor set
Arithmetic unit
104
updating section B'
Main controller Jp
26
Gate way
5
Pre processing section Ak
4
Pre processing section At
3
Pre processing section A1
2
Rivoting Section

Fig. 9

Fig.10


start
n-[n/k]k>0
Yes
No
p=0
Parallel pre Processing A1
Parallel pre Processing A2
Parallel pre processing Ak
Parallel updating B'
Post elimination processing C
p=[n/k]-1
Yes
No
p=p+1
p=0
Parallel pre Processing A1
Parallel pre Processing An-[n/k]k
p=[n/k]
Yes
No
p=p+1
Patrallel pre Processing Ak
Parallel updating B'
Post elimination processing C
p=[n/k]
Yes
No
end

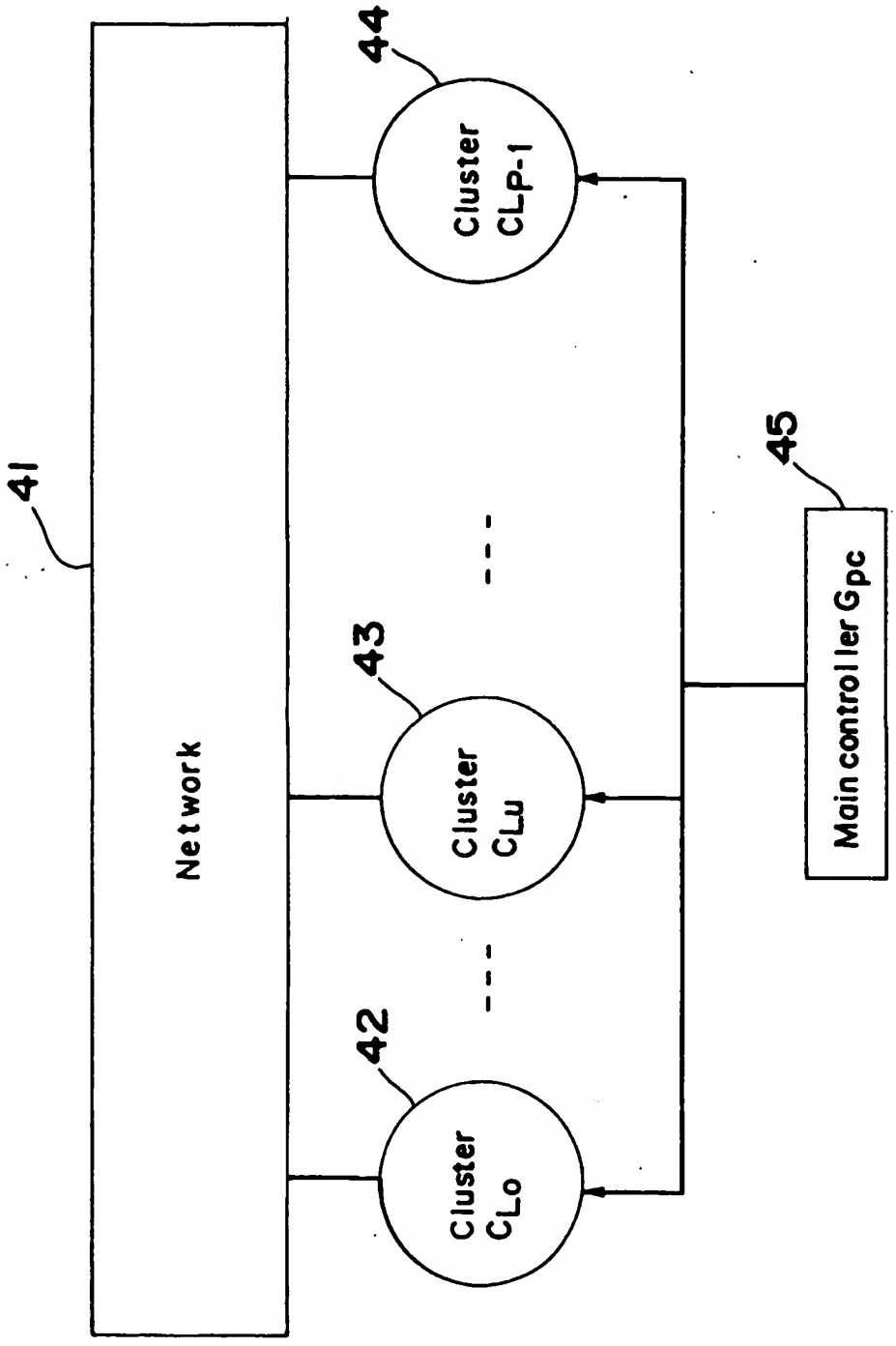

41
Network
44
Cluster
CLP-1
43
Cluster
CLu
42
Cluster
CLo
45
Main controller Gpc

Fig.11

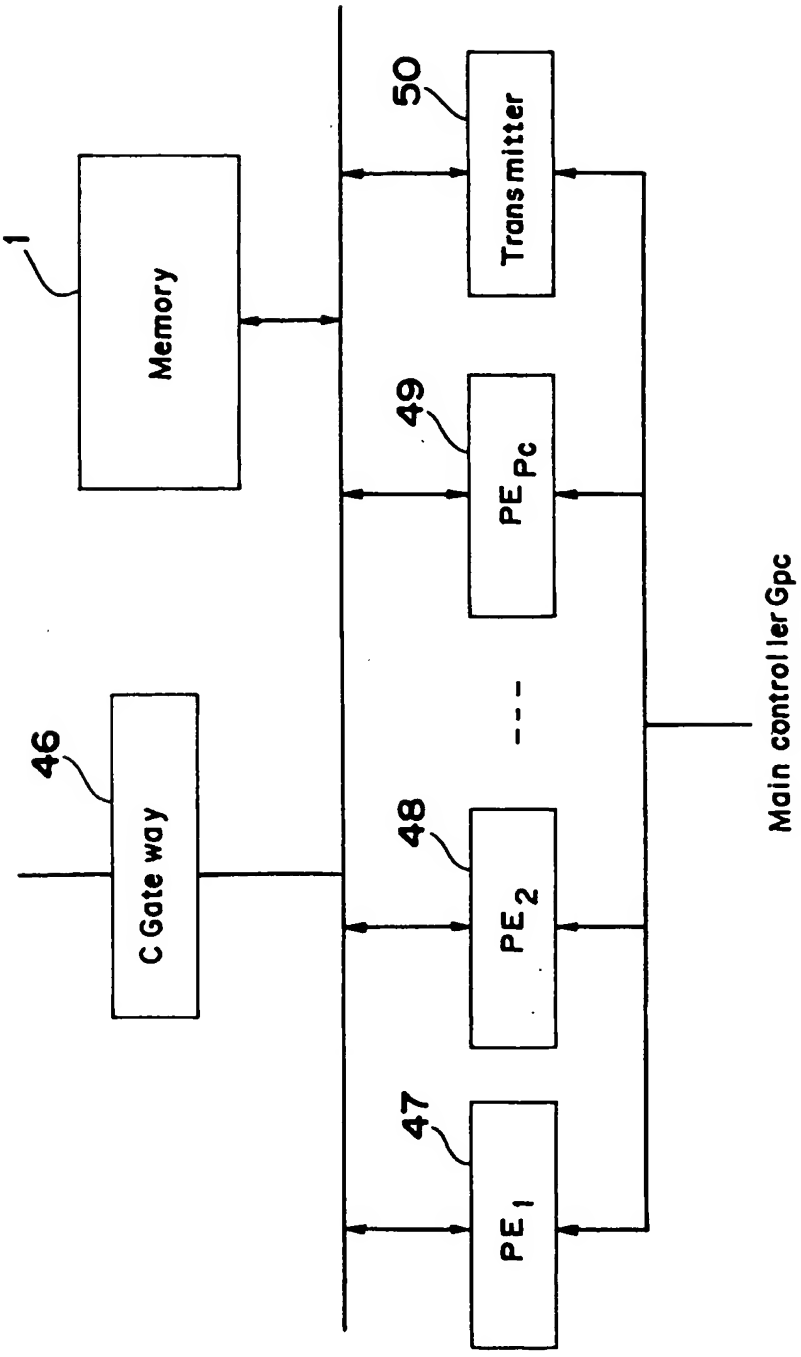

1
Memory
50
Transmitter
49
PE Pc
46
C Gate way
48
PE 2
47
PE 1
Main controller Gpc

Fig.12

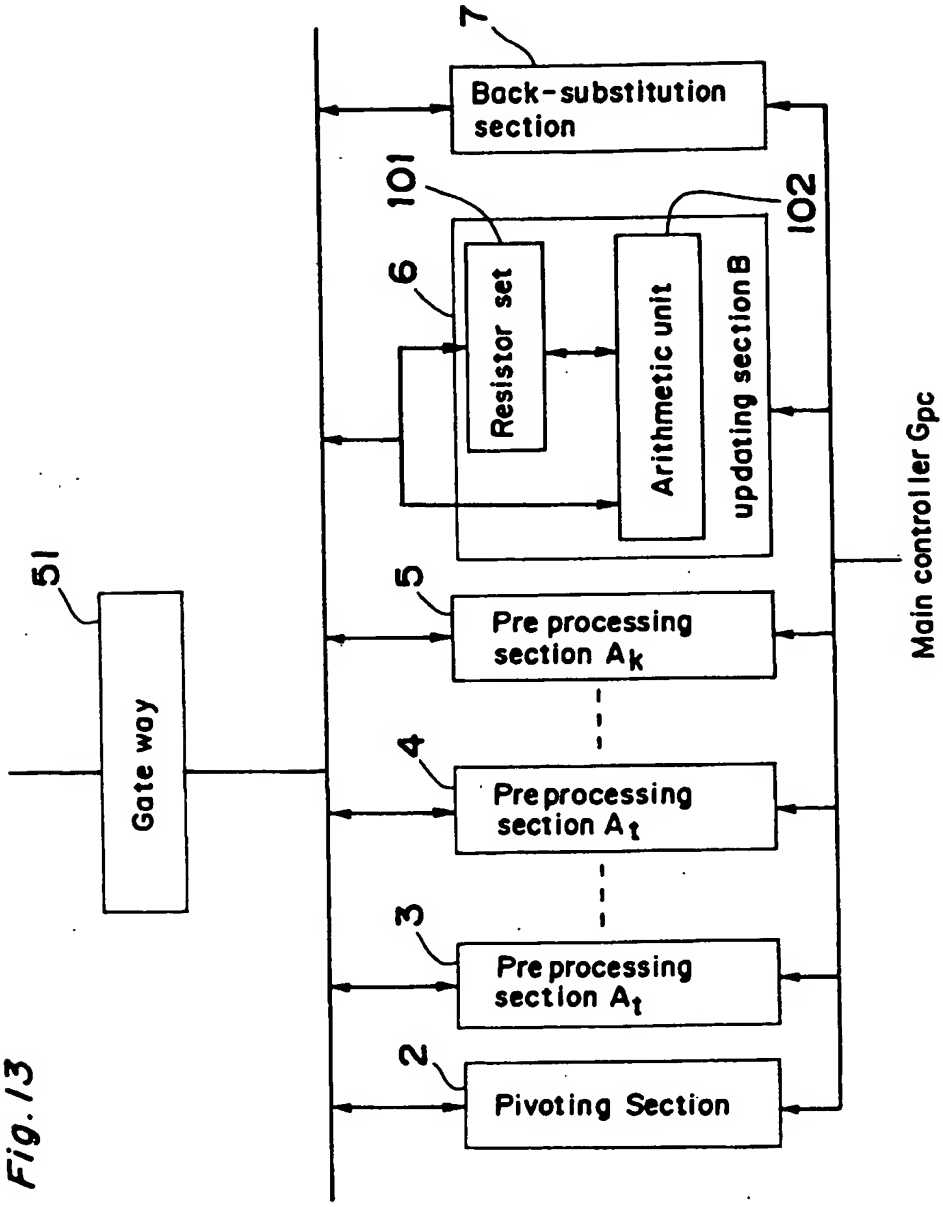
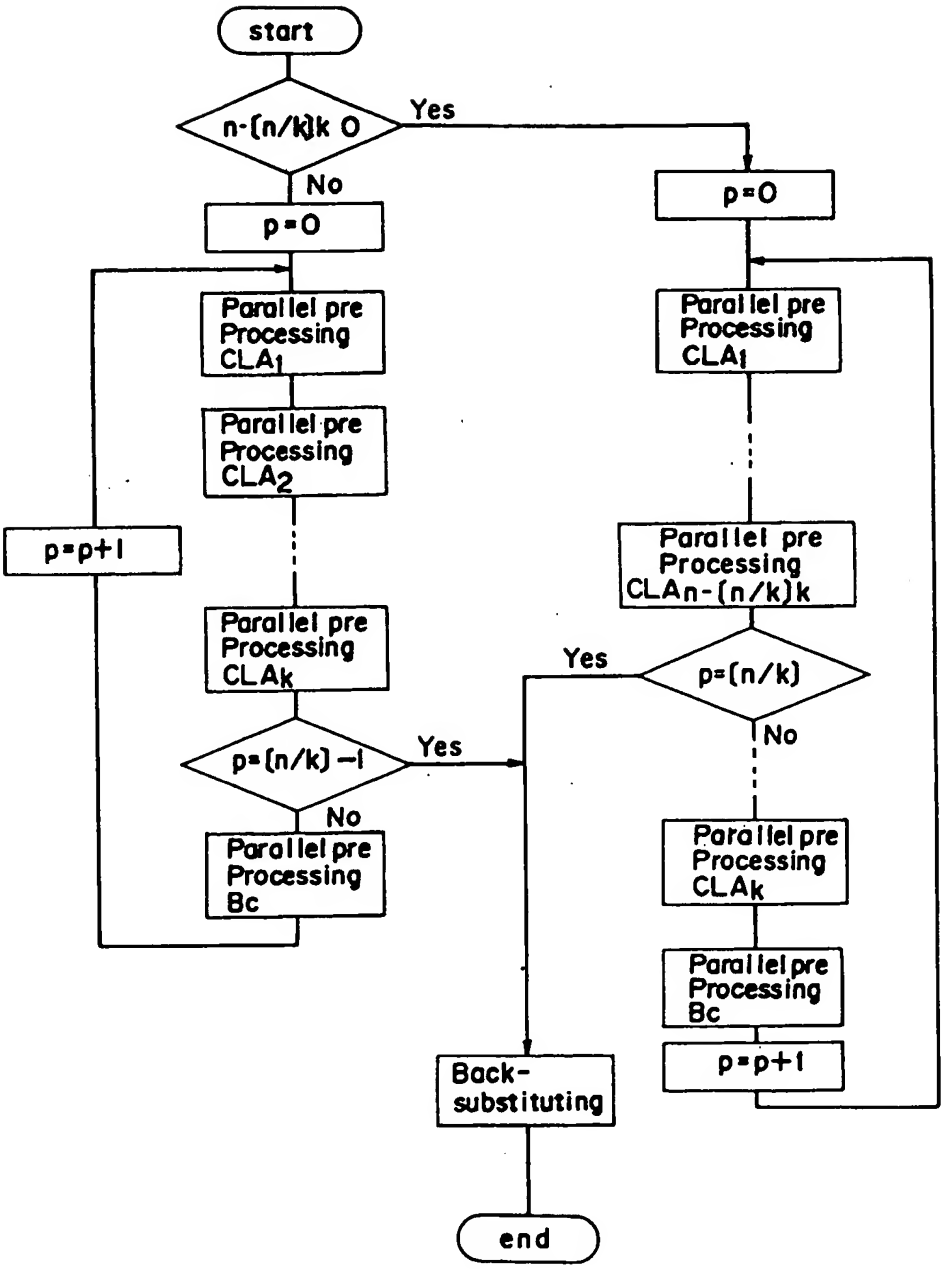

7
Back-substitution section
101
102
6
Resistor set
Arithmetic unit
updating section B
51
Gate way
5
Pre processing section A_k
4
Pre processing section A_t
3
Pre processing section A_t
2
Pivoting Section
Main controller Gpc

Fig. 13

Fig.14


start
n-(n/k)k 0
Yes
No
p=0
p=0
Parallel pre Processing CLA1
Parallel pre Processing CLA2
Parallel pre Processing CLAk
p=p+1
p=(n/k)-1
Yes
No
Parallel pre Processing Bc
Parallel pre Processing CLA1
Parallel pre Processing CLAn-(n/k)k
Yes
p=(n/k)
No
Parallel pre Processing CLAk
Parallel pre Processing Bc
p=p+1
Back-substituting
end

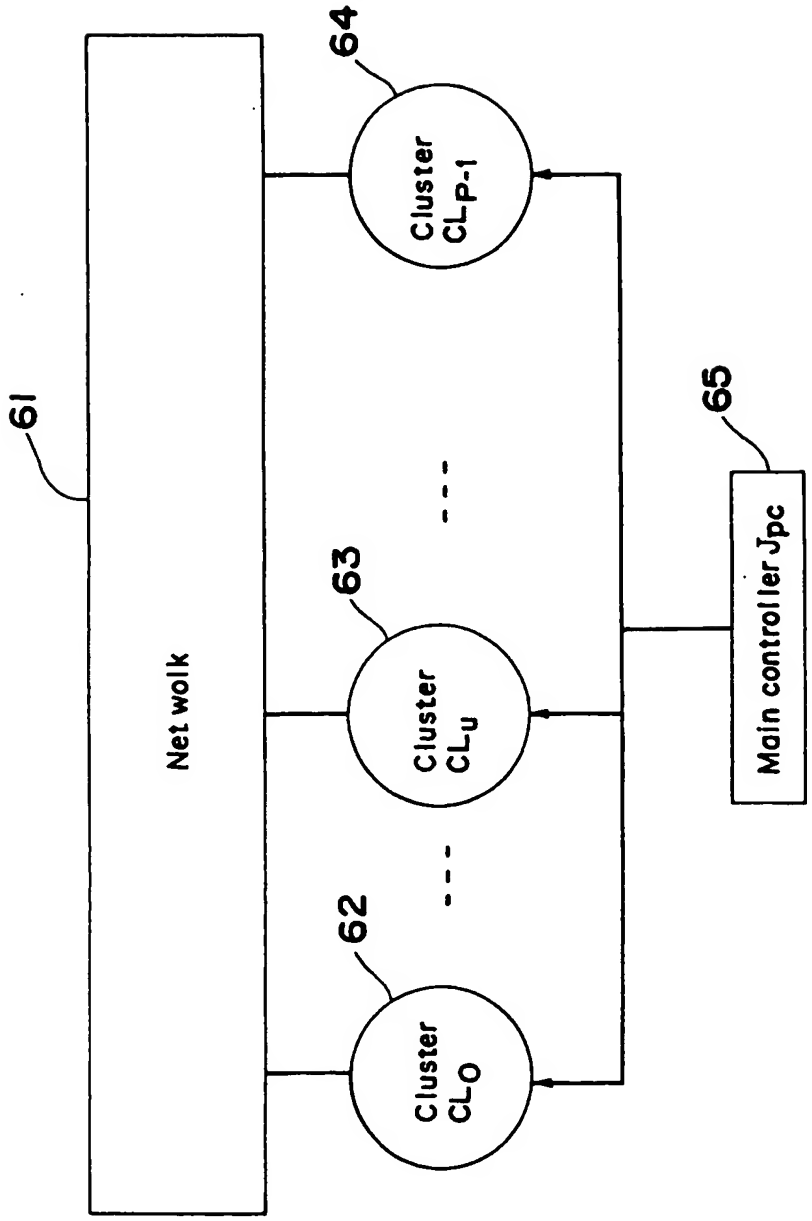

61
Net wolk
62
Cluster
CL0
63
Cluster
CLu
64
Cluster
CLP-1
65
Main controller Jpc

Fig.15

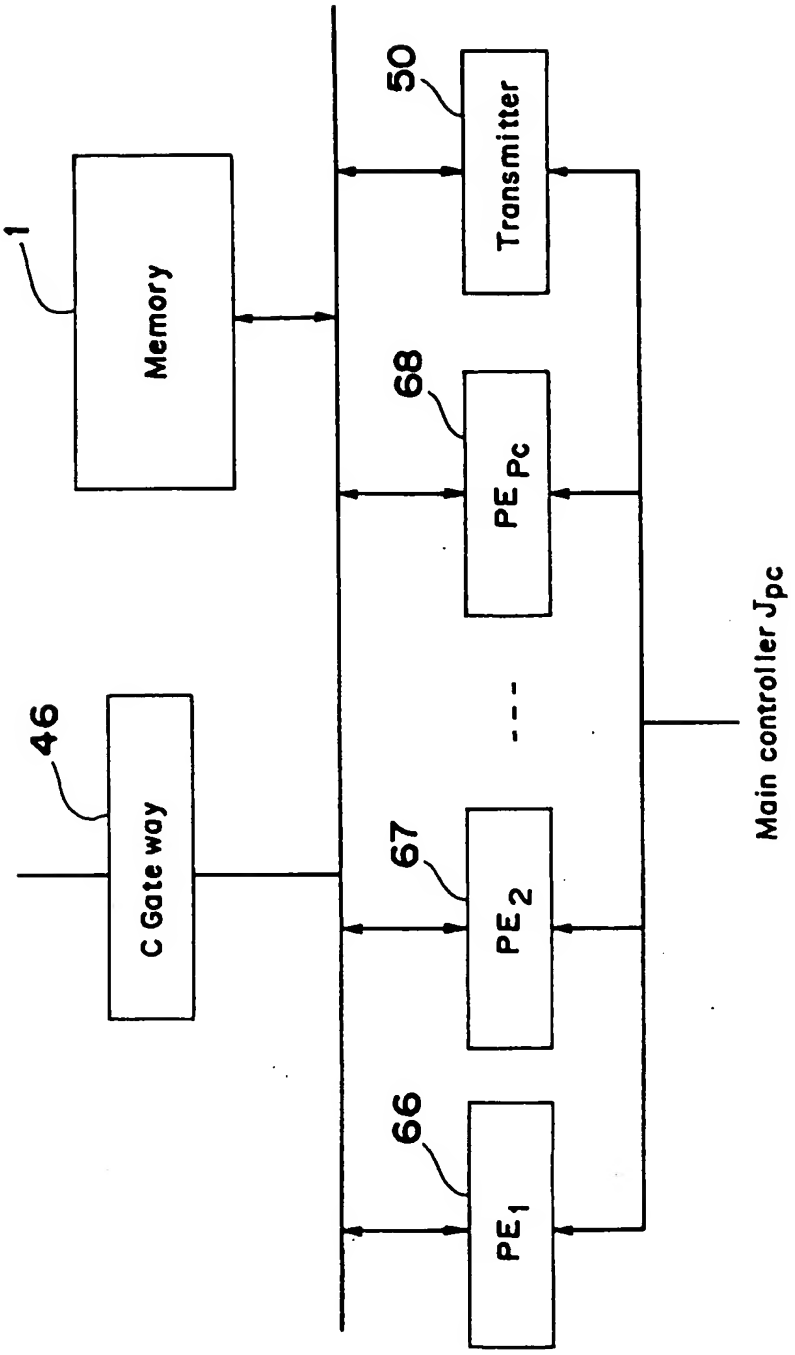

1
Memory
50
Transmitter
68
PE Pc
46
C Gate way
67
PE 2
66
PE 1
Main controller Jpc

Fig. 16

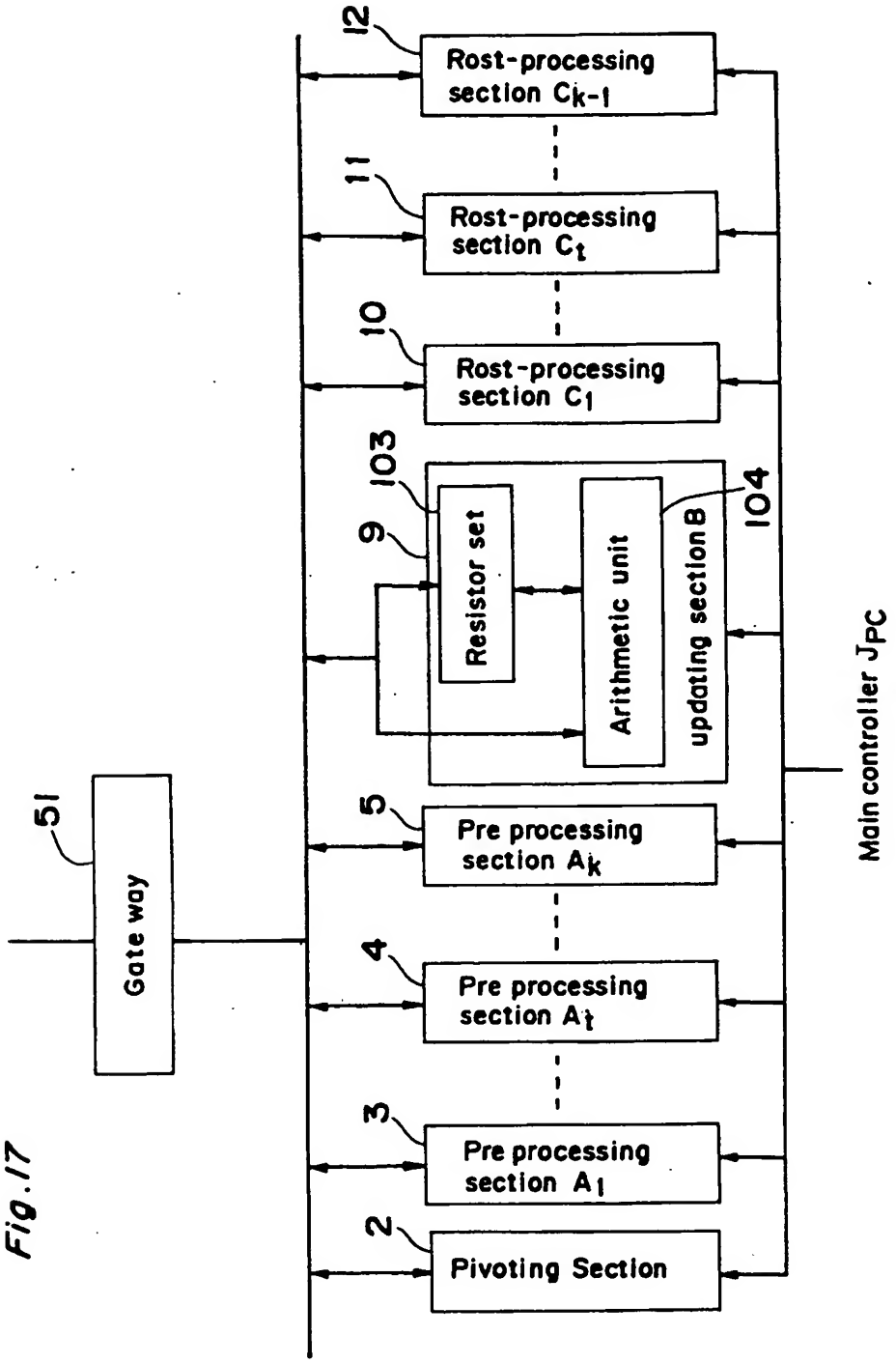

12
Rost-processing section C_{k-1}
11
Rost-processing section C_t
10
Rost-processing section C_1
103
9
Resistor set
Arithmetic unit
104
updating section B
Main controller J_{PC}
51
Gate way
5
Pre processing section A_k
4
Pre processing section A_t
3
Pre processing section A_1
2
Pivoting Section

Fig.17

Fig. 18

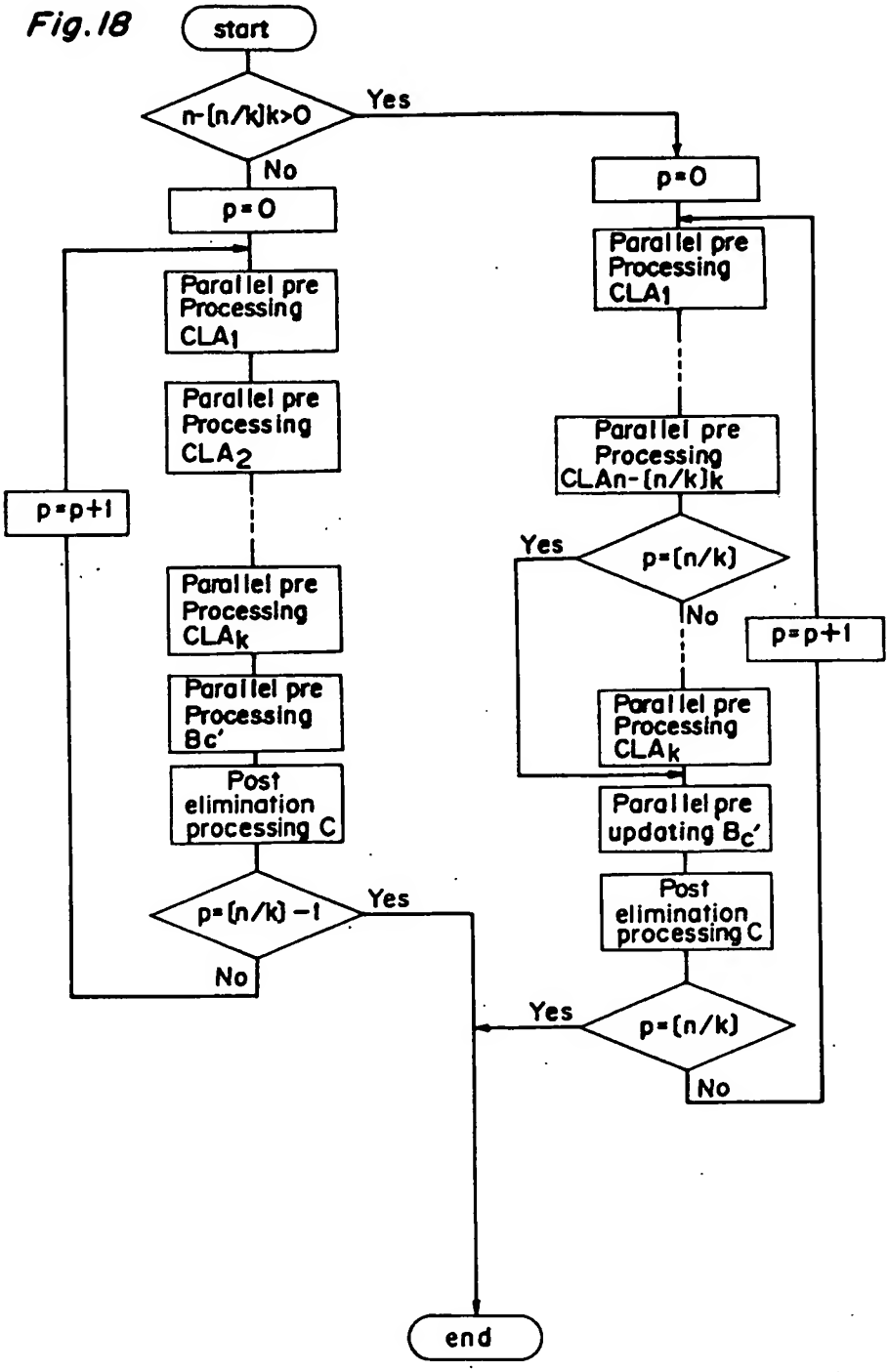

start
n-(n/k)k>0
Yes
No
p=0
Parallel pre Processing CLA1
Parallel pre Processing CLA2
Parallel pre Processing CLAk
Parallel pre Processing Bc'
Post elimination processing C
p=(n/k)-1
Yes
No
p=p+1
p=0
Parallel pre Processing CLA1
Parallel pre Processing CLAn-(n/k)k
Yes
p=(n/k)
No
Parallel pre Processing CLAk
Parallel pre updating Bc'
Post elimination processing C
Yes
p=(n/k)
No
p=p+1
end

Fig.19

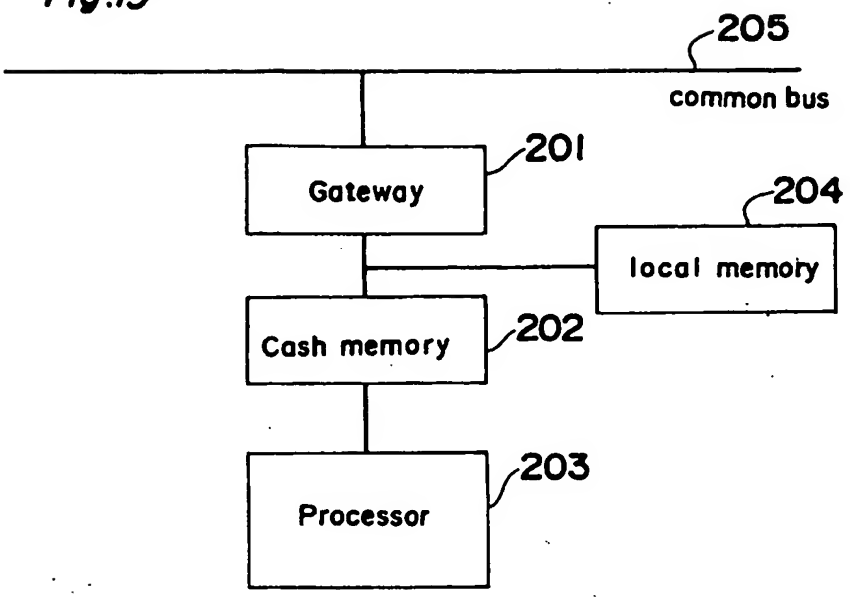

205
common bus
201
Gateway
204
local memory
Cash memory
202
Processor
203

Fig.20

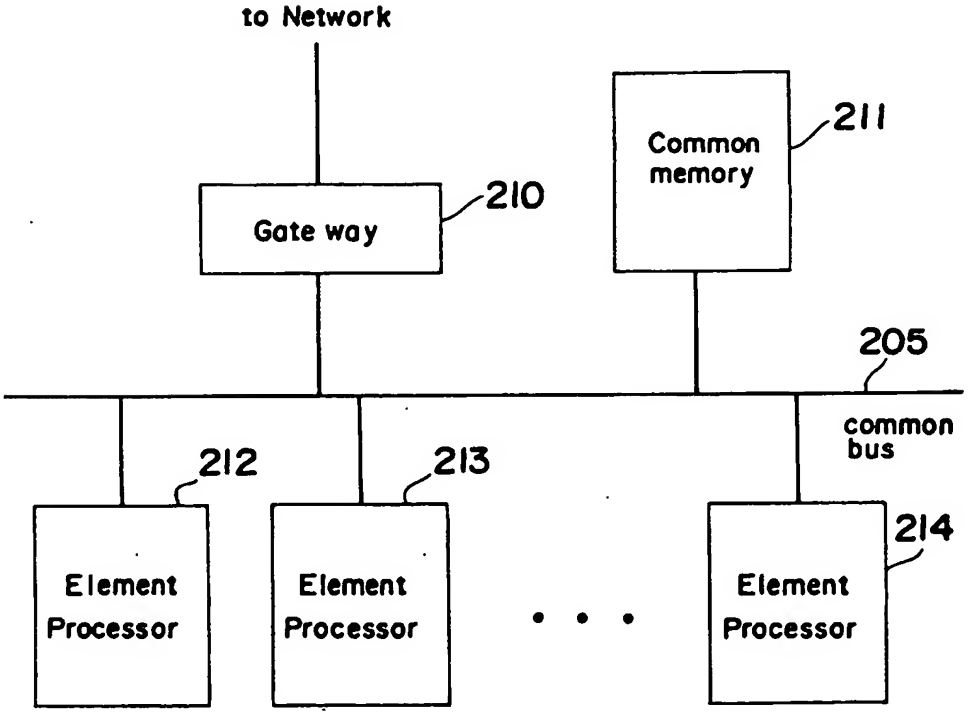

to Network
Common memory
211
Gate way
210
205
common bus
212
213
214
Element Processor
Element Processor
Element Processor

Fig.21

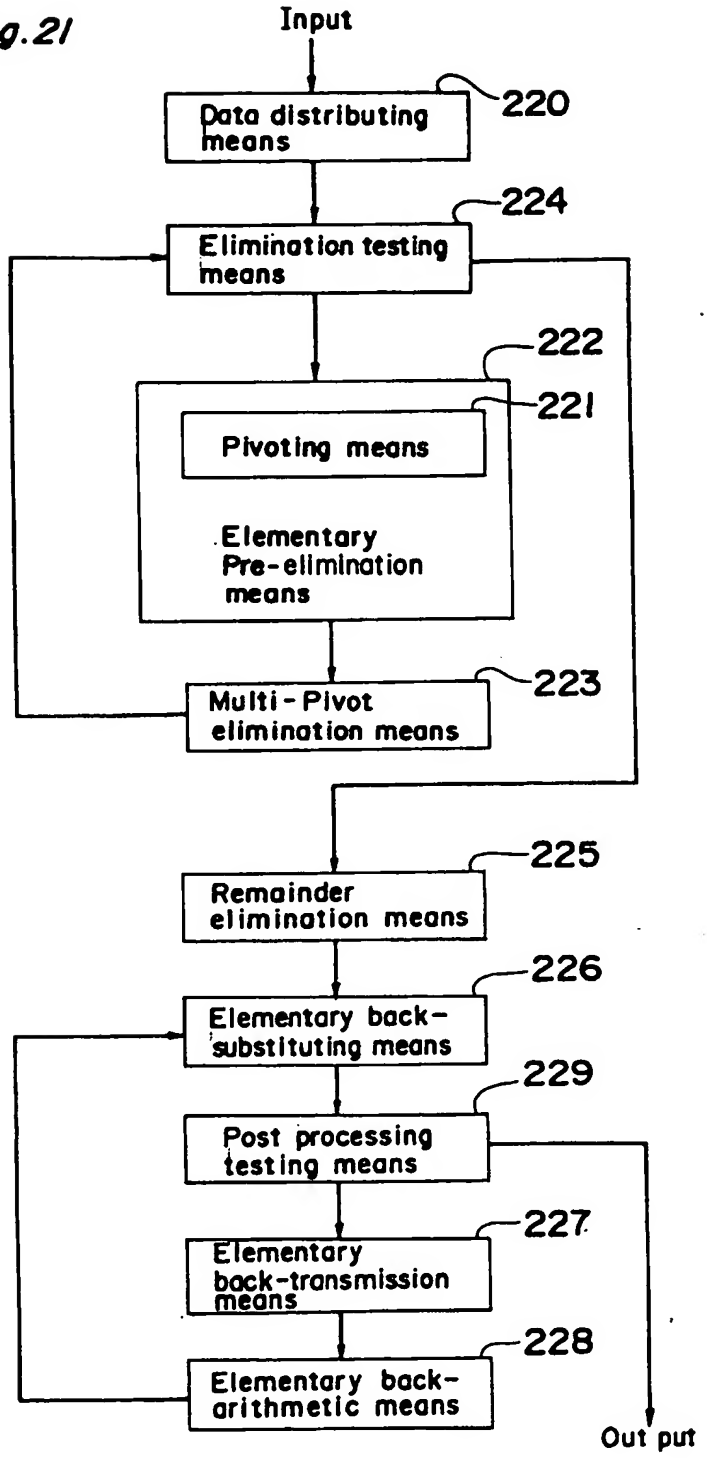

Input
Data distributing means
220
Elimination testing means
224
222
221
Pivoting means
Elementary Pre-elimination means
Multi-Pivot elimination means
223
Remainder elimination means
225
Elementary back-substituting means
226
Post processing testing means
229
Elementary back-transmission means
227
Elementary back-arithmetic means
228
Out put

Fig.22

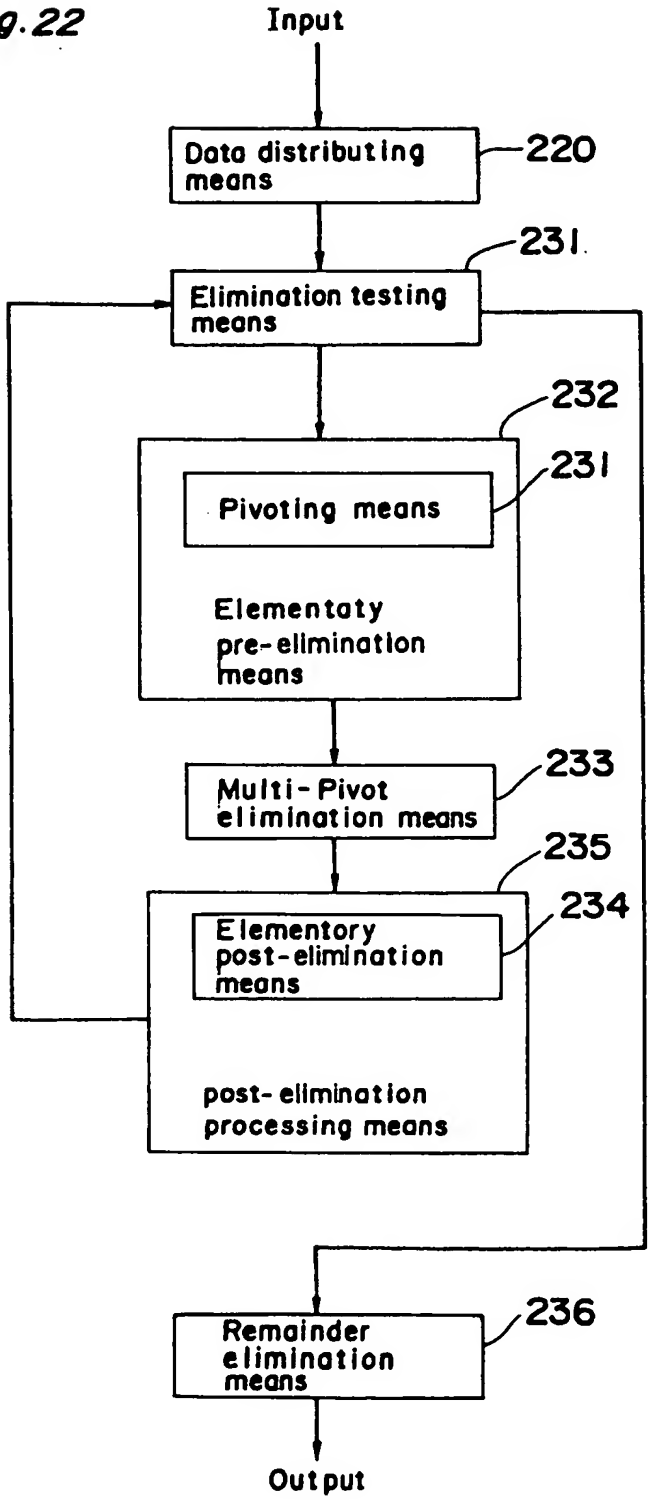
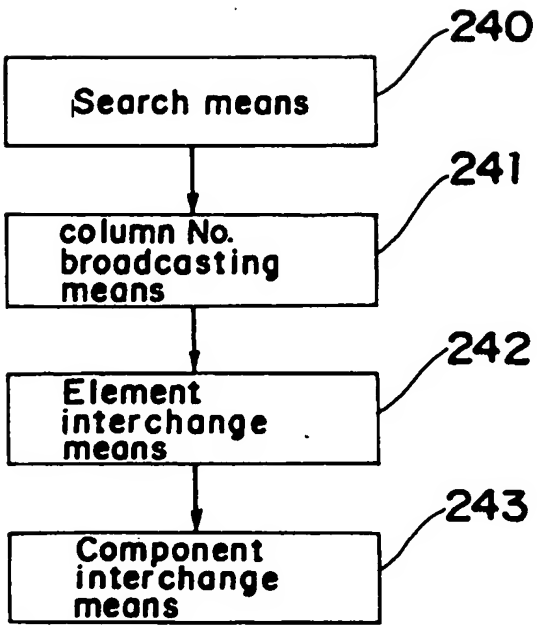

Input
Data distributing means
220
231
Elimination testing means
232
231
Pivoting means
Elementaty pre-elimination means
233
Multi-Pivot elimination means
235
234
Elementory post-elimination means
post-elimination processing means
236
Remainder elimination means
Output

Fig. 23


240
Search means
241
column No. broadcasting means
242
Element interchange means
243
Component interchange means

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 523 544 A3**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **92111660.4**

(51) Int. Cl.5: **G06F 15/324**, G06F 15/347

(22) Date of filing: **09.07.92**

(30) Priority: **12.07.91 JP 172168/91**
15.07.91 JP 173616/91

(43) Date of publication of application:
20.01.93 Bulletin 93/03

(84) Designated Contracting States:
DE FR GB

(88) Date of deferred publication of the search report:
10.08.94 Bulletin 94/32

(71) Applicant: **MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.**
1006, Oaza Kadoma
Kadoma-shi, Osaka-fu, 571(JP)

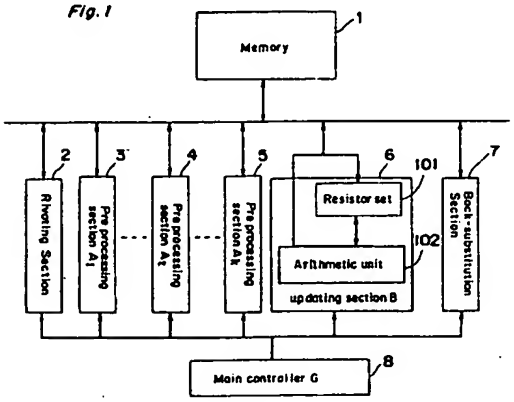
(72) Inventor: **Mochizuki, Yoshiyuki**
3-9-3 Sagisu,
Fukushima-ku
Osaka-shi, Osaka(JP)

(74) Representative: **Elsenführ, Speiser & Partner**
Martinistrasse 24
D-28195 Bremen (DE)

(54) Calculating equipment for solving systems of linear equation.

(57) A linear calculating equipment comprises a memory for storing a coefficient matrix, a known vector and an unknown vector of a given system of linear equations, a pivoting device for choosing pivots of the matrix, a plurality of preprocessors for executing K steps of preprocessing for multi-pivot simultaneous elimination, an updating device for updating the elements of the matrix and the components of the vectors, a register set for storing values of the variables, a back-substitution device for obtaining a solution and a main controller for controlling the linear calculating equipment as a whole.

Fig. 1



EP 0 523 544 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 92 11 1660

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.5)
A	TRANSACTIONS OF THE INSTITUTE OF ELECTRONICS, INFORMATION AND COMMUNICATION ENGINEERS OF JAPAN vol. 72, no. 12 , December 1989 , TOKYO JP pages 1336 - 1343 XP000100267 NOBUYUKI TANAKA ET AL 'SPECIAL PARALLEL MACHINE FOR LU DECOMPOSITION OF A LARGE SCALE CIRCUIT MATRIX AND ITS PERFORMANCE' * page 1336, right column, line 35 - page 1337, left column, line 28 * * page 1339, left column, line 1 - page 1340, left column, line 20 * * page 1340, right column, line 13 - page 1341, left column, line 13 * ---	1-10	G06F15/324 G06F15/347
A	PARALLEL COMPUTING vol. 11, no. 4 , August 1989 , AMSTERDAM NL pages 201 - 221 XP000047174 GITA ALAGHBAND 'PARALLEL PIVOTING COMBINED WITH PARALLEL REDUCTION AND FILL-IN CONTROL' * page 202, line 30 - page 204, line 13 * * page 207, line 4 - page 208, line 4 * ---	1-10	TECHNICAL FIELDS SEARCHED (Int.Cl.5) G06F
A	IEEE TRANSACTIONS ON COMPUTERS vol. 32, no. 12 , December 1983 , NEW YORK US pages 1109 - 1117 MANDAYAM A. SRINIVAS 'OPTIMAL PARALLEL SCHEDULING OF GAUSSIAN ELIMINATION DAG'S' * page 1110, left column, line 3 - page 1111, left column, line 6 * ---	1-10	
	-/--		

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	6 June 1994	Barba, M

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 92 11 1660

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.5)
A	PARALLEL COMPUTING vol. 13, no. 3 , March 1990 , AMSTERDAM NL pages 289 - 294 XP000140487 HYOUNG JOONG KIM ET AL 'A PARALLEL ALGORITHM SOLVING TRIDIAGONAL TOEPLITZ LINEAR SYSTEM' * page 291, line 6 - page 293, line 10 * ---	1-10	
A	PROCEEDINGS OF THE 1989 POWER INDUSTRY COMPUTER APPLICATION CONFERENCE, IEEE PRESS NEW YORK US 1 May 1989 , SEATTLE, WASHINGTON, US pages 9 - 15 XP000091286 D.C.YU ET AL 'A NEW APPROACH FOR THE FORWARD AND BACKWARD SUBSTITUTION OF PARALLEL SOLUTION OF SPARSE LINEAR EQUATIONS BASED ON DATAFLOW ARCHITECTURE' * page 9, right column, line 6 - line 16 * * page 9, right column, line 39 - page 10, left column, line 3 * * page 10, left column, line 21 - page 11, left column, line 20 * ---	1-10	TECHNICAL FIELDS SEARCHED (Int.Cl.5)
A	PROCCEDINGS OF THE IEEE 1983 INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, IEEE PRESS NEW YORK US vol. 1/3 , 2 May 1983 , NEWPORT BEACH, CALIFORNIA US pages 214 - 217 R.M.KIECKHAFER ET AL 'A CLUSTERED PROCESSOR ARRAY FOR THE SOLUTION OF THE UNSTRUCTURED SPARSE MATRIX EQUATIONS' * page 214, right column, line 35 - page 215, right column, line 31 * * page 216, right column, line 10 - line 49 * --- -/--	1-10	

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	6 June 1994	Barba, M

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 92 11 1660

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.5)
A	PROCCEDINGS OF THE 1986 IBM EUROPE INSTITUTE SEMINAR ON PARALLEL COMPUTING, NORTH HOLLAND AMSTERDAM NL 11 August 1986 , OBERLECH, AUSTRIA pages 99 - 106 XP000014765 IAIN S. DUFF 'PARALLELISM IN SPARSE MATRICES' * page 100, line 1 - line 33 * * page 101, line 27 - page 103, line 17 * -----	1-10	

TECHNICAL FIELDS SEARCHED (Int.Cl.5)

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	6 June 1994	Barba, M

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)